LATTICE SIMULATIONS OF PHASE MORPHOLOGY IN MODEL MEMBRANE SYSTEMS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

In Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Jonathan Jacob Amazon

January 2014

LATTICE SIMULATIONS OF PHASE MORPHOLOGY IN MODEL MEMBRANE SYSTEMS

Jonathan Jacob Amazon, Ph.D.

Cornell University 2014

Model membrane systems are a useful tool for studying the role of lateral phase separation in relation to the lipid rafts on living cells. One key aspect of the phase separation observed in both living and model systems is phase morphology, the size and shape of phase domains. For many lipid systems the phase morphology tends towards a single large round (macroscopic) domain to minimize the perimeter to area ratio. By contrast, several lipid systems have been found to have nanoscopic (nanodomains) or modulated (periodic and thermodynamically stable) phase domains. The explanation for the large excess in phase boundary present in these mixtures requires the use of a competing interactions model, in which line tension (energy per unit length) competes with curvature and/or electrostatics to stabilize non-trivial phase morphologies.

To study these interactions in a way that accurately represents the membrane shape we present and implement a lattice model for use in computer simulations of phase morphology. These simulations show that on a spherical surface, curvature can compete with line tension to produce modulated phases that closely match the size and characteristics observed on giant unilamellar vesicles (GUV). The model is extended to include electrostatic repulsion, which is found to break up macroscopic domains on large unilamellar vesicles (LUV) into irregular clusters with correlation lengths consistent with size measurements of nanodomains.

BIOGRAPHICAL SKETCH

Jonathan Amazon grew up surrounded by the sun, beaches, and vibrant night life of Miami Beach, Florida. He spent most of his time there ignoring this, and instead found interest in math, science, drawing, music, Lego, and computers. Growing up, Jon was not always good with technical subjects. After almost failing algebra I, Jon moved on to study music and trumpet playing at a performing arts high school. It was here that an off-hand remark from his Algebra II teacher prompted Jon to read about the strange history behind the number zero, and kindled his personal interest in math and physics.

After high school, Jon headed north to Rensselaer Polytechnic Institute to pursue a degree in physics and math. It was here that he made many good friends and had his first encounter with snow, which he immediately regretted. This sudden realization that being cold is bad motivated Jon to work hard and finish his degree in just three years. This did not solve the cold problem however, as Jon moved to Cornell University to begin graduate school work on a PhD in Biophysics. During his time in graduate school Jon worked in Jerry Feigenson's lab performing simulations of phase morphology observed on lipid vesicles. His project went through many failed iterations and crashing simulations, but eventually succeeded with the appearance of modulated phases stabilized by curvature.

With the completion of his dissertation, Jon plans to start a post-doc research job at HHMI Janelia Farm working with Dr. Shaul Druckmann. Here he will be less cold and continue the development of theoretical models that describe the behavior of neural networks. Jon's ultimate goal is to one day be able to explain (if even at the shallowest level) the nature and existence of intelligence in an otherwise mechanistic universe.

To my parents, Kip and Barbara Amazon.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my PI and advisor, Dr. Gerald Feigenson, for the opportunity to work with him in this lab. Jerry's enthusiasm and guidance have helped me stay motivated while here at Cornell. Throughout my project I have met with dead-end calculations, re-invented the wheel (over and over), and espoused outlandish ideas. All the while, Jerry has been patient and helpful, pulling me back to reality, but still leaving me the creative freedom to fail or succeed on my terms. He has helped me grow both as a professional and as a person. Thank you Jerry.

A big thanks to all of my professors and committee members for helping me learn many new and useful skills and for advice and discussions about my projects; Dr. Bruce Land, Dr. Earl Kirkland, Dr. Warren Zipfel, Dr. Michelle Wang, Dr. James Sethna, and Nate Ellis. During my time in the Feigenson lab, I have had the pleasure of working with many friends and colleagues who have all been instrumental in the progression of this work; Dr. Fred Heberle, Dr. Shih Lin Goh, Dr. Robin Petruzielo, Dr. Nelson Morales-Penningston, Dr. Jing Wu, Tatyana Konyakhina, and David Ackerman (keep on keepin' on). For research funding I thank the NIH Molecular Biology Training Grant 1-T32 GM 08267 and the NSF MCB 0842839 research grant awarded to Gerald W. Feigenson.

My friends and family, who have been supportive of me throughout these years. My father, Kip, my mother, Barbara, my sister, Jackie, and my nieces and nephews, have all been there to help me relax and unwind during my short visits home, so that I could return refreshed and ready to work. My friends Zac Petit, Steve Herring, Eric Dzienkowski, and Orlando Scalia. A special thanks to my girlfriend, Tiffany (love you). Her suggestion, over two years ago, to "Just add more zeros!" were the exact words that led to the successful simulation of modulated phases and the subject of this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Thesis Outline

Lipid phase behavior in bilayer membranes has proven to be an important aspect of biochemistry. This work aims to model and explain the size and shape of phase domains using a coarse grained lattice simulation that emulates the material properties of the membrane. A lattice simulation is a highly coarse grained approach that enables scalability from nanoscopic structures all the way up to optical phenomena with minimal change in computation time. This model is implemented in a custom simulation environment and used to explore the effects of various competing interactions, such as line tension, curvature, and electrostatic repulsion, on phase morphology. The results of these simulations are compared to experimental observations and used to make predictions of phase morphology when model parameters are varied.

A key motivation for this modeling approach is the appearance of modulated phases in the four-component system DSPC/DOPC/POPC/chol (chapter 2). This was a surprising observation that provided a key constraint for model parameters and insight into the microscopic interactions that stabilize nanodomains. To better understand this from a theoretical

perspective a competing interactions model is proposed that studies phase sorting by curvature and line tension on a deformable surface (section 2.3.2). The model is implemented in a simulation that uses a Monte Carlo relaxation algorithm on a triangulated surface to find a minimal free energy state (section 2.4.6). The key questions that we address using this simulation approach are summarized below.

Can competition between curvature and line tension explain modulated phases? With appropriate parameters we are able to predict the existence of modulated phases on a phase separated GUV (section 2.5.1). Furthermore, it is found that varying only line tension is sufficient to reproduce the macro-modulated-nanoscopic transition observed in GUV studies [36](whereas reasonable variations of other parameters were not able to capture this behavior).

Are modulated phases thermodynamically stable or kinetically trapped? The modulated phase patterns arising from this model were found to be thermodynamically stable (section 2.5.2). The details of the phase morphology were found to be directly influenced by model parameters such as phase fraction, line tension, and bending moduli (section 2.5.3).

Do modulated phases appear without the background curvature and topological constraints present on a spherical membrane? This question is addressed by constructing a flat sheet lattice (section 3.3.2) and performing simulations with parameter sets that give rise to modulated phases. It is found that modulated phases do not persist on a flat membrane where background curvature is absent (section 3.4.2).

The competing interaction model with line tension and curvature is found to explain modulated phases. The natural extension of that finding is to ask if *nanoscopic* phase separation that is observed in some model lipid systems may be similarly explained by a

competing interactions model. One interaction to consider at the small scale is electrostatic repulsion between permanent molecular dipoles. In appendix B it is shown that electrostatic interactions are very short ranged, on the order of a few nanometers. This is still a few lipid shells, and thus the competition of line tension with dipole repulsion of individual lipids must be considered.

Can electrostatics break up macroscopic domains at sub optical scales? Using the lattice simulation it is found that electrostatics can break up macroscopic domains and produce nanoscopic phase separation under salt concentrations that mimic physiological conditions. These results are summarized in chapter 3 as well as the effects of changing the magnitude of the dipole interaction (section 3.4.3) and the decay length within the bilayer plane (section 3.4.3).

While comparing nanoscale and GUV-scale phase morphology using this model it was found that coarse-graining has a profound effect on the energetic parameters, which impairs our ability to meaningfully compare simulations at different coarse-grain levels. This was due to line tension being a running coupling constant, a parameter that must change with coarse-graining to account for the loss of degrees of freedom. This problem was addressed using renormalization group techniques (section 3.4.1). Once renormalization was taken into account, comparisons could be made between the parameters used to study modulated phases and those used to study nanodomains.

Before each of these topics is presented and discussed in greater detail a brief introduction to lipid structure and phases is given (section 1.2). Modulated phases and the experimental motivation for their discovery are discussed (section 1.2.1). To motivate the study of model systems the lipid raft hypothesis is presented, which bridges the gap between lipid phase behavior and biological function (section 1.2.2). The current state of computational modeling

of lipid systems and the many different techniques that are employed is summarized (section 1.3) with emphasis on the lattice modeling approach employed in this study (section 1.3.2).

Technical details and more minor results are given in the appendices. The supplemental material for chapters 2 and 3 are given in appendix A. Our exploration of the Poisson-Boltzmann equation to better understand the electrostatic environment of the membrane is outlined in appendix B. The algorithms for constructing the unbiased spherical lattices used in the simulation are given in appendix C. A potential model for interpreting phosphorescence lifetime measurements is outlined in appendix D. The various instruments and techniques developed in my time in the Feigenson lab are summarized in appendix E. The full documentation of the lattice simulation implementation is given in appendix F which includes a "read me" for future users (section F.1), an explanation of the program flow/algorithms (section F.1), a listing of all data structures used (section F.2), and the file formats used by the simulation (section F.3).

1.2 Lipid Chemistry and Phases

Amphiphilic lipids are a class of molecules important to the structure and function of living cells. The eukaryotic cell contains hundreds of different lipid species that are involved in a variety of processes such as energy storage, metabolism, signaling, transport, and are a major structural component of the membrane. The lipids of interest to most plasma membrane research can be classified into three groups: **sterols**, which are small conjugated hydrocarbons with a small polar head group; **sphingolipids**, which have a sphingosine backbone, a fatty acid residue in amide linkage, and a polar head group; and **phospholipids**, which have a glycerol backbone, two fatty acyl residues, and a polar head group.

Each lipid species is distinguished by the chemical structure of its fatty acid residues (number of carbons and degree of saturation) and its head group. The structures of some

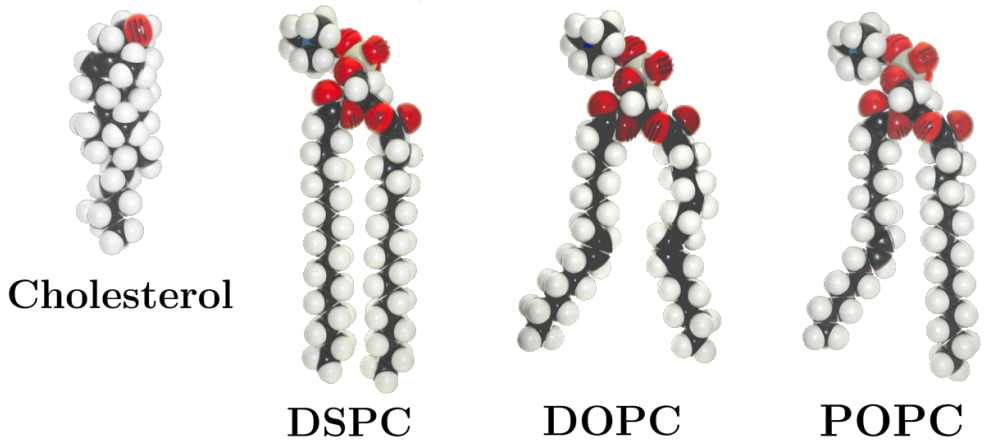


Figure 1.1: Space filling structure of lipid molecules considered in this study. From left to right; **Cholesterol**, a sterol abundant in animal cell membranes that is important in the formation of the L_o phase; **DSPC**, a phospholipid with two saturated chains that pack well with cholesterol; **DOPC**, a phospholipid with two mono-unsaturated chains that partitions favorably into an L_d phase; **POPC**, a hybrid lipid with one saturated and one unsaturated chain that also partitions favorably into the L_d phase.

of the lipids used in this study are shown in figure 1.1. From left to right are cholesterol (chol), the most abundant sterol in animal cell membranes; distearoylphosphatidylcholine (DSPC), with two saturated acyl chains; dioleoylphosphatidylcholine (DOPC), with two mono-unsaturated acyl chains; and palmitoyl,oleoyl-phosphatidylcholine (POPC), a hybrid lipid, having one saturated and one unsaturated chain. Each of the phospholipids shown has a choline head group, which is abundant in the plasma membrane. The different lengths of fatty acid chains and their degree of saturation affect the order and mixing properties of the lipids, which has proven important to their biological function.

1.2.1 Phase Behavior of Lipids

The amphiphilic nature of detergent and lipid molecules drives the formation of a variety of structured, hydrated phases [96]. Some of these phases are shown in figure 1.2, with the particular phase details dependent on the lipid type and structure. The formation of these

hydrated phases is driven by the tendency to minimize exposed hydrophobic surface, which is achieved by the polar headgroups shielding the fatty acyl chains from the surrounding water. Consequently the ratio of cross-sectional areas of the headgroup and the hydrophobic moiety is of key importance in determining the phase behvaior. Most glycerophospholipids exist as monomers at concentrations below 10^{-12}M, but readily form the lamellar (L_d) bilayer phase at higher concentrations. At high temperatures or when the lipids have small headgroups, such as phosphatidylethanolamine (PE), the inverse hexagonal (H_{II}) phase is formed. Under appropriate conditions phopholipids may also form a monolayer, which is a membrane comprised of a single sheet of lipid molecules.

Detergents form micelles to shield their small (1-chain) hydrophobic moiety with comparatively large headgroups. When the detergent concentration gets high enough that the headgroups cannot be fully hydrated, the hexagonal (H_I) phase forms to more efficiently pack the lipids. Reverse micelles can be formed when lipids are dissolved in a non-polar solvent and it is the polar headgroups that must be sequestered from the solvent. There are also more exotic bicontinuous phases that can be formed which partition the aqueous medium into two continuous but distinct chambers. Except for the lamellar phase, the other structures have limited relevance to biology, but monolayer membranes play an important role in respiration, and detergent micelles and bicontinuous cubic phases have proven useful in purifying and crystallizing membrane proteins [64, 95].

The lamellar phase is the lipid structure of biomembranes. The semi-permeable lipid bilayer serves to compartmentalize the cell into organelles, form spherical liposomes for trafficking, and (in the case of the plasma membrane) enclose the entire volume of the cytosol. The bilayer also serves an important biochemical role. The flexibility of the bilayer permits the formation of an arbitrarily high surface area within an enclosed volume, which is ideal for biochemical processes known be catalyzed at the membrane surface (such as at the inner mitochondiral membrane). An array of techniques are used to create and study

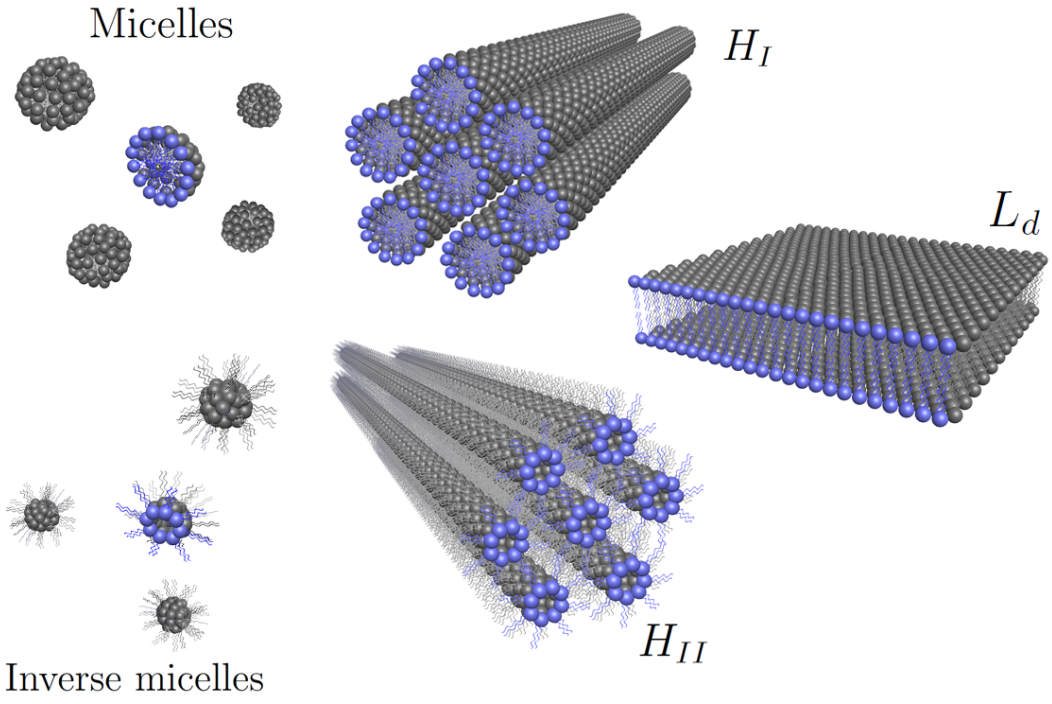


Figure 1.2: Lipids form various structured phases in water. The structures shown are: micelles; hexagonal phase H_I; lamellar phase L_d; inverse hexagonal phase H_{II}; and inverse micelles.

lipid bilayers. The most common technique used by most researchers is to produce liposomes from dry lipid films. Gentle hydration [1] and electroswelling [7, 112, 86] are used to produce giant unilamellar vesicles (GUV) that can be observed using optical microscopy. Rapid solvent exchange (RSE) produces stacks of bilayers [18, 17] that can be used for spectroscopic measurements or extruded to produce large unilamellar vesicles (LUV).

Bilayer Phases

The different lipid species that compose the membrane have different head groups, acyl chain lengths, and degrees of saturation, which affect their in-plane mixing behavior. The tendency for certain lipid species to aggregate/repel influences the formation of different bilayer phases [31, 63]. A cross section of these phases is shown in figure1.3. The liquid-disordered phase (L_d) is characterized by a low degree of chain conformational order, low positional order, and high rate of diffusion ($D_d \approx 10^{-8}$ cm^2/s) [3]. By contrast, the gel phase (L_β) is found to have high chain conformational order, high positional order, and significantly slower diffusion constant ($D_\beta \approx 10^{-16}$ cm^2/s).

For model systems to mimic the plasma membrane, cholesterol must be included as a major component. Cholesterol has special mixing properties with the surrounding phospholipids; in some lipid systems it precipitates a cholesterol rich liquid-ordered phase (L_o). This phase is primarily composed of saturated lipids and cholesterol, and found to have high chain conformational order, low positional order, and a diffusion constant comparable to that of the L_d phase ($D_o \approx 10^{-9}$ cm^2/s). This cholesterol-rich phase is thought to be an integral component of the plasma membrane.

Simplified model systems containing as few as three lipid components have been found to capture a very rich composition-dependent phase behavior. A ternary lipid mixture composed of a low-T_m lipid (L_d at physiological temperatures), a high-T_m lipid (L_β at physiological temperatures), and cholesterol produce phase diagrams that exhibit all three phases

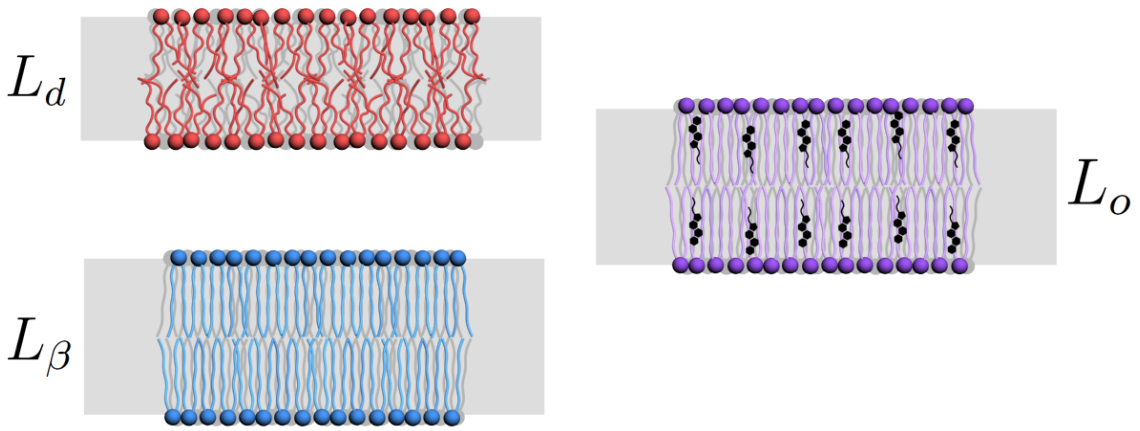


Figure 1.3: Three lamellar phases: Liquid disordered phase L_d, characterized by low chain order and high diffusion coefficient; Gel phase L_β, characterized by high chain order and low diffusion coefficient; and Liquid ordered phase L_o, characterized by high chain order, high diffusion coefficient, and enrichment of cholesterol (small black molecules).

(L_d,L_β, and L_o) with regions of phase coexistence [33, 125]. A typical ternary phase diagram that has these regions of phase coexistence is shown in figure 1.4. The liquid-liquid ($L_d + L_o$) coexistence region in the center is of keen interest, as this may be a manifestation of the lateral heterogeneities known to exist in the plasma membrane [42].

There are many ways that phase diagrams are solved for lipid systems. When phase separation occurs at optical scales, fluorescent dyes and GUVs can be used to observe the phase separation using optical fluorescence microscopy [33, 125, 63, 98, 62]. This technique sometimes gives acceptable boundary determination and compositional control, however one must be mindful of light induced artifacts from fluorophore photochemistry [126, 9]. When phase separation is below optical resolution, a combination of Forster resonance energy transfer (FRET) [16, 19, 30, 35, 123] and electron spin resonance (ESR) [83] can be used to determine phase boundaries [43, 33]. When a gel-fluid transition is present, phase boundaries can be well-determined using differential scanning calorimetry (DSC) [119, 83].

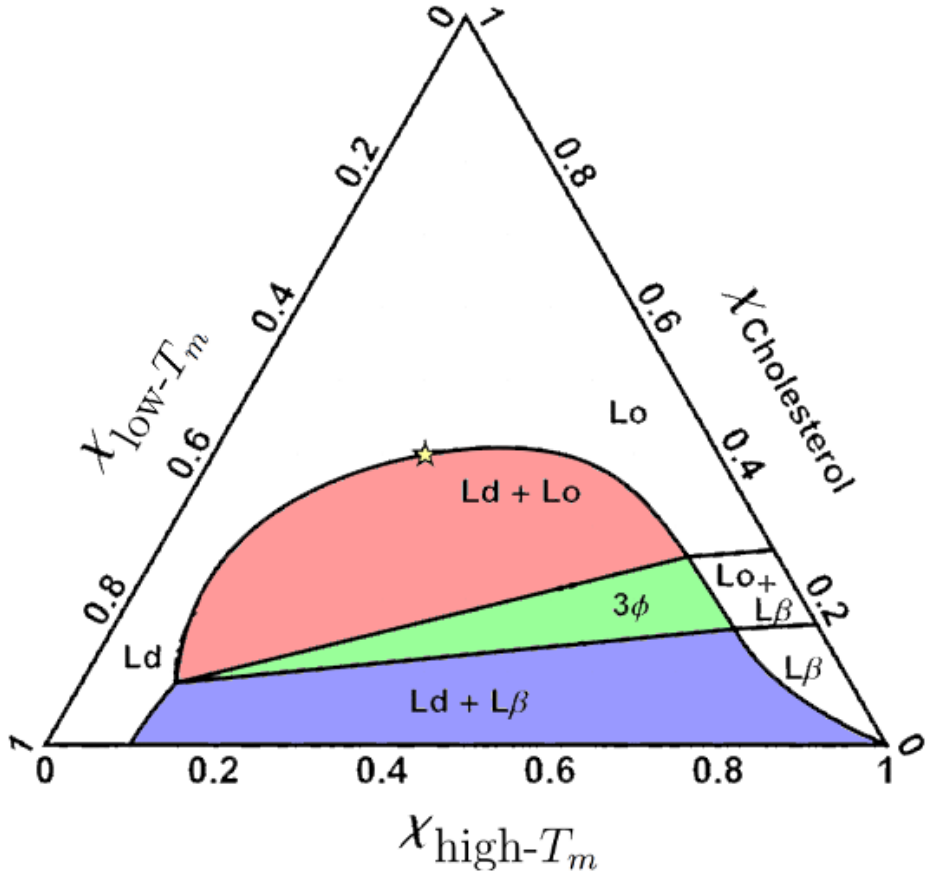


Figure 1.4: Example of a ternary phase diagram for a low-T_m lipid, high-T_m lipid, and cholesterol. The major regions of interest are the two-phase liquid-gel (L_d+L_β) coexistence region (blue), the three-phase (L_d+L_o+L_β) region (green), and the liquid-liquid (L_d+L_o) coexistence region (red). (Image modified and used with permission of Fred Heberle [43].)

Phase Morphology and Nanodomains

The discovery of coexisting phases in model lipid mixtures was a huge step forward for our understanding of biomembranes. The next step lies in our understanding of phase morphology, which is the arrangement, shape, and size of phase domains. A simple model for phase morphology is to assume that a phase-separated system minimizes the phase boundary, as given by the following Hamiltonian,

$$\mathcal{H}_{\text{line}} = \gamma L \tag{1.1}$$

where γ (line tension) is an energy per unit length and L is the total phase boundary. The minimization of this energy forms a single round domain, which has the highest area-to-perimeter ratio. This phase morphology is observed in particular model mixtures [125], and is referred to as macroscopic phase domains.

Macroscopic phase domains are convenient for determination of phase boundaries and for the measurement of some energetic parameters [124, 97, 82], but they are not the typical phase morphology observed on GUVs made with natural phospholipids. In most lipid systems, typified by DSPC/POPC/chol, SM/POPC/chol, and DPPC/DLPC/chol, the GUVs appear uniform by optical microscopy. When these systems are examined using FRET, ESR, and small angle neutron scattering (SANS) they reveal the presence of nanodomains, phase separation at the nanometer scale [43, 45, 83, 33, 111]. While the experimental evidence of nanodomains is well established, the theoretical explanation for their existence remains in question. The great excess of interface created when domains are broken up on the nanometer scale necessitates a more complicated energy landscape than equation 1.1.

Few explanations for the existence of nanodomains have been proposed. One explanation is to treat nanodomains as critical fluctuations near a miscibility critical point [48, 71, 118]. This model presumes the membrane composition to be in a one phase region of the phase

diagram, but close enough to the critical point to produce transient fluctuations on the order of the correlation length. Another model treats nanodomains as a microemulsion in a one-phase region, with hybrid lipids acting as lineactant molecules [94, 79]. Both of these models explain the uniform appearance of GUVs, but do not account for the existence of a well-defined two-phase region, evidenced by high-resolution FRET surfaces and the upper boundary of the three-phase region terminating as a straight line [43].

If nanoscopic mixtures are taken to be in a two-phase region of the phase diagram then a competing interactions model explains the stabilization of nanodomains [100]. Curvature is one possible competing interaction with line tension, whereby mechanical deformations of the membrane incur an energy penalty [38, 46, 11]. This model can capture kinetic stabilization of small domains by an effective curvature-mediated repulsion [117, 23] and, if the bending moduli of the two phases are sufficiently different, thermodynamically stable domains via frustration/buckling of the membrane [49, 106, 5]. Another interaction that could potentially stabilize nanodomains is electrostatic repulsion between adjacent lipid molecules due to their permanent molecular dipoles [24, 120, 85, 121, 88]. This repulsion affects the stability of large domains, forcing them to disperse to minimize this frustration [116, 4, 56].

Modulated Phases

To better understand the interactions that dictate the size and stability of nanodomains a four-component model system can be used to gradually transition from a macroscopically phase-separated to a nanoscopically separated mixture [61, 36, 62, 5]. By starting in the two-phase region of the macroscopic mixture DSPC/DOPC/chol, and gradually replacing DOPC with POPC, the two-phase region of the nanoscopic system DSPC/POPC/chol is formed. We define the replacement ratio ρ as,

$$\rho = \frac{[\text{DOPC}]}{[\text{DOPC}]+[\text{POPC}]} \tag{1.2}$$

in this way, $\rho = 0$ corresponds to a nanodomain forming mixture and $\rho = 1$ corresponds to a macroscopic domain-forming mixture.

It was expected that domain size would gradually increase from $\rho = 0$ to $\rho = 1$, however the experimental observations were much more surprising. Instead of a gradual change in size or an abrupt change from nano to macro, the system underwent a narrow window ($0.15 < \rho < 0.25$) of modulated phases. **Modulated phases** are patterns of L_d and L_o phase with well-defined periodicity and thermodynamic stability. These patterns resemble stripes or honeycomb like arrangements of circular domains with nearly uniform size. Similar patterns had been observed before in model membranes subjected to additional interactions [38, 57, 91].

Using a competing interactions model, simulations showed that curvature and line tension explain the appearance and stability of modulated phases [5, 36]. These simulations found that varying line tension γ as a monotonically increasing function of ρ produced results in remarkable agreement with experimental observations. This relationship between γ and ρ makes measurements of the ρ-window a useful proxy for line tension, which is notoriously difficult to measure [115, 29, 97]. By determining if the ρ-window has shifted to higher/lower values of ρ under controlled perturbations (addition of protein or other lipid components) we may identify this shift with a decrease/increase of line tension. Similar modulated phase transitions have also been observed in other four-component systems that exhibit both macroscopic and nanoscopic phase behavior [44].

1.2.2 Lipid Rafts

The interest in membrane phase morphology, specifically nanodomains, is driven by its potential relevance to the membranes of living cells [32, 28, 104]. Lateral heterogeneities on the animal cell plasma membrane have been implicated in a number of important cellular processes [101, 99, 25] such as trafficking [6, 55], signal recognition [37, 103], and viral assembly/budding on the inner leaflet (though evidence of phase separation has proven elusive on inner leaflet models [122]). When observed using optical microscopy, the surface of living cells does not display the characteristic phase separation visible in certain model lipid mixtures, but there is evidence of membrane lateral heterogeneity at sub-optical scales [42]. The identification of these structures with nanoscopic L_o phase domains is one form of the lipid raft model [69, 13, 27, 84]. Figure 1.5 shows a simplified representation of lipid rafts and proteins on the plasma membrane.

The role of lipid composition was not always considered an integral part of the cell's biochemistry. The fluid mosaic model of the membrane [105] was an important conceptual step in understanding the organization of the plasma membrane. It describes the membrane as a lipid bilayer in which proteins are associated and embedded. The proteins are then free to diffuse within the two dimensional plane of the bilayer to form complexes and initiate biochemical processes. The fluid mosaic model treats the lipid component of the plasma membrane as a purely structural feature, where the biochemistry of the membrane is driven primarily by proteins. This interpretation brings up an important question; given that just one lipid species is sufficient to produce a lipid bilayer, why would the cell tolerate the metabolic cost of maintaining and regulating the hundreds of lipid species that make up the plasma membrane? Furthermore, the cell is found to actively maintain asymmetric lipid compositions between the two leaflets of the plasma membrane; a state that has proven difficult to reproduce in experimental systems [20]. These observations indicate that the

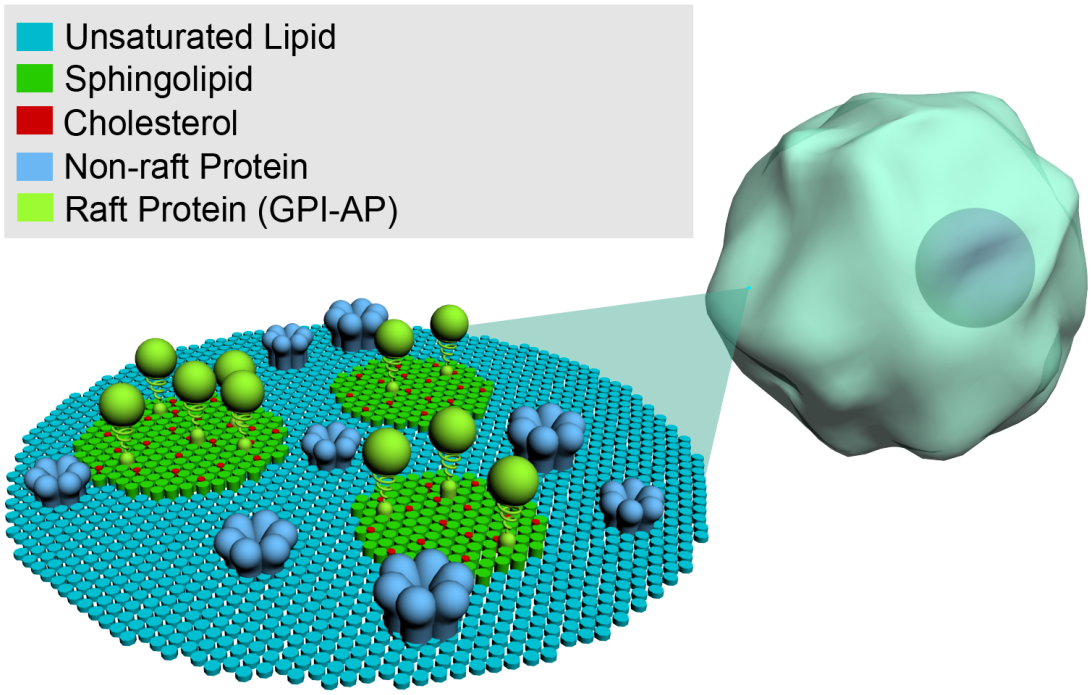


Figure 1.5: Representative section of the plasma membrane showing raft-mediated lateral organization. Certain proteins partition into raft regions of the membrane to increase the efficiency of protein-protein interactions important for many cellular processes.

lipid composition of the plasma membrane and its leaflets is of key importance to the cells function and organization.

Evidence of lateral organization on the plasma membrane has been around for decades [110, 55]. The observation that epithelial cells have polarized apical and basolateral membranes prompted the association of sphingolipid-sterol interactions with membrane function [102]. One important but indirect line of evidence of L_o like regions of the plasma membrane came from the discovery of detergent resistant membranes (DRM). When plasma membrane is dissolved in 1% Tritton-X100 at 4^o C, portions of the membrane that are enriched in sphingolipid and cholesterol remain intact [27]. Another important experiment found that chemical crosslinking of GPI-anchored proteins, known to be associated with membrane rafts, enabled the visualization of rafts as optically resolvable domains [41]. These two experiments reveal sterol-dependent organization of the membrane, but have come under scrutiny for their

use of substantial temperature and chemical perturbations [68, 13, 81, 67]. Less perturbative means of isolating rafts have been successfully used to study their composition and properties. Increasing membrane tension (osmotic swelling) at physiological temperature and pH together with crosslinking can coalesce rafts into large optically resolvable domains [10, 2]. Diffusion measurements, made on giant plasma-membrane vesicles (GPMV) using floures-cently labeled molecules, can also detect anomalous diffusion behavior, thought to be caused by diffusion in and out of membrane rafts [65, 77].

One aspect of the lipid raft model that is pivotal to its biological relevance is the sorting of proteins into raft regions of the membrane. Some proteins present on the surface of the plasma membrane, such as GPI-anchored proteins, are raftophilic (preferentially partition into the raft phase). The distribution of proteins into rafts has been studied using fluorescent labels to quantify clustering (found to be at the nanometer scale) [66] and raft affinity deduced by protein extraction from DRMs [14]. There is also evidence that cross-linking of membrane proteins can drive the formation of phase domains on the membrane [40, 34]. Even so, the identification of proteins that partition strongly into the L_o phase in model systems has seen limited success.

The exact structure, function, and lifetime of lipid rafts remain controversial [76], but with continuing advances in super-resolution techniques [93] and less perturbative methods of studying the true state of living cell membranes, lipid rafts will continue to be an essential conceptual tool for interpreting new findings.

1.3 Modeling Lipid Membranes

Mathematical modeling of biological systems is a rapidly progressing area of research that draws heavily on such varied disciplines as physics, math, chemistry, biology, and computer science. It is used to test hypotheses and make predictions in systems that are too delicate to

be experimentally analyzed or too complicated to be treated analytically. This makes modeling invaluable to all branches of biology. In the context of modeling lipid membranes and its associated proteins there are two main paradigms; atomic-level **molecular dynamics**, which captures detailed information about every atom in a small number of molecules, and **coarse grained methods**, which use approximate models to capture large scale behavior.

1.3.1 Molecular Dynamics

Molecular dynamics (MD) simulations are a class of simulation packages that model a small number of molecules at atomistic resolution. The two most prominent molecular dynamics packages in use now are Gromacs (**GRO**ningen **MA**chine for **C**hemical **S**imulations), which is an open source package developed at the University of Groningen, and CHARMM (**C**hemistry at **HAR**vard **M**acromolecular **M**echanics), which is a licensed package developed at Harvard University.

Both packages solve Newton's laws of motion and can keep track of the positions and momentum of every single atom. This is done by defining a potential energy in terms of the positions of all of the atoms and then using leapfrog integration to evolve the system forward in time. The potential energy of each atom is composed of two types of terms. The bonded terms, which include torsion angles, dihedral angles, and bond lengths; and the non-bonded terms, which include electrostatic interactions, and Van der Waals forces (typically a 6-12 potential).

Molecular dynamics has been successful in answering questions about the details of lipid order and structure such as the stability of cholesterol clusters [21], the free energy of interactions between lipid molecules [12], the overall orientation of the zwitterionic head groups [107], and the electrostatic environment in and around the membrane [78]. The drawbacks of the MD approach are that the time/spatial scales necessary to understand

lipid phase behavior are orders of magnitude above what current MD simulations are capable of. For an 'all atom' MD simulation the maximum bilayer size is roughly 1024 lipids (512 in each leaflet). This corresponds to a box size of only 20×20 nm^2, which is the estimated size of a single nanodomain [45]. In addition, the time scales realizable for such a simulation are on the order of 100nS, which is approximately the time it would take for a lipid to diffuse about 1nm [3]. This is not sufficient to study the equilibrium lateral organization of lipids in mixtures in the bilayer.

1.3.2 Coarse Graining Methods

To study the large scale ensemble behavior of lipid membranes a more computationally friendly approach is to consider a coarse grained model. A coarse grained model is a mathematical framework in which degrees of freedom are integrated into a single entity, thus vastly reducing the complexity (and hence computational time) of the simulation. This allows coarse grained simulations to probe virtually any temporal or spatial scale, so long as an appropriate graining is chosen. Unlike all atom MD simulations, coarse grained models are highly specialized to answer questions about specific systems, so there are limited coarse graining packages available for lipids. Coarse grained approaches are usually outlined algorithmically and the implementation is left to individual researchers. Here we will introduce some of the more common coarse grained models that have seen success in applications of lipid phase behavior and understanding the plasma membrane.

Coarse Grain Molecular Dynamics

To bridge the gap between the microscopic detail of all atom MD and the high level approximations of the Ising/lattice models there, are a few coarse grained versions of MD that are commonly used. The most subtle of these is known as United Atom (UA), which is a force

field included with the **GROMACS** package. The UA force field treats some hydrogen atoms that are bound to carbon atoms (such as methyl and methylene groups in the acyl chains) as a single atom with special parameters. This simplification greatly reduces the number of atoms in the simulation and hence allows for a marked speed-up of production runs.

The next level of graining, and perhaps the most used in MD, is the MARTINI force field. The MARTINI model approximates groups of atoms as a single spherical bead that is given properties to emulate the underlying structure. This reduces a lipid from having hundreds of atoms to evolve per time step to, for example, just twelve 'MARTINI atoms.' This provides a huge speed up, which enables the study of much larger systems even up to entire small vesicles [72]. The MARTINI force field has been used to study the partitioning of lipids at an interface [92], the effects of adding proteins to the bilayer [109], and until recently, has been the only dynamics simulation to yield phase separation on simulation time scales [22, 73].

The Cooke model, where lipids can be represented by as few as three beads (headgroup and two hydrocarbon beads), can be used to study even larger scale behavior. This type of coarse graining has been used to study the effects of adding curvature inducing peptides to the bilayer [90] and to measure the saddle-splay modulus [50]. Coarse graining further than this is usually not necessary as one bead per lipid can more effectively be treated by other models.

Monte-Carlo Methods

In any modeling of a statistical ensemble we will at some point have to draw a random number. This leads us to a general class of algorithms known as Monte-Carlo methods. Monte-Carlo methods rely on using random numbers to sample a distribution or state space and enable us to make measurements of the statistical properties of the model being studied. Nearly every discipline of science, math, and engineering (and even some forms of art [89])

use Monte-Carlo methods to solve a dizzying array of problems.

The power of Monte Carlo methods is in the ability to handle huge numbers of degrees of freedom with relative ease. This is useful for statistical systems because most relevant quantities (average energy, heat capacity, correlation lengths, etc.) can be expressed as an integral over a high dimensional state space. Many of the coarse graining methods that will be outlined in this section use a Monte Carlo method to evolve the simulation through its state space. One common algorithm that is used is the Metropolis-Hastings algorithm, which evolves the simulation in a way consistent with the free energy landscape (captures both enthalpic and entropic contributions) [75]. Each perturbation is accepted or rejected based on the change in energy, ΔE, in the Hamiltonian. If the energy is found to increase (an unfavorable perturbation) then it is accepted according to a Boltzmann probability, $e^{-\Delta E/kT}$. Kinetic trapping is a problem in Monte Carlo simulations with a complicated Hamiltonian, so most techniques use simulated annealing to avoid local energy minima [106].

The Ising Model

The Ising model is a versatile framework for treating such varied physical phenomena as magnetic metals, gases adsorbed to surfaces, liquid crystals, neural networks, and lipid membranes. It captures important thermodynamic features of these systems such as phase transitions and critical phenomena, which make it an ideal framework for studying lipid phase behavior.

The standard Ising model assumes a fixed lattice (such as a 2D square lattice) where each position, i, represents a magnetic spin (σ_i) which may take one of two states, spin up ($\sigma_i = 1$), or spin down ($\sigma_i = -1$). The Hamiltonian assumes that each spin interacts only with its neighboring spins and contributes to the total energy according to,

$$\mathcal{H}_{\text{Ising}} = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j. \tag{1.3}$$

Here we have used the notation $\langle \cdot, \cdot \rangle$ to restrict the sum to terms where i and j are neighboring lattice positions. The coupling constant, J, dictates the strength of the interactions and hence the energetic contributions to the Hamiltonian. We can see that when $J > 0$, we would want $\sigma_i = \sigma_j$ to minimize the energy (known as the ferromagnetic Ising model) and when $J < 0$ we would want $\sigma_i = -\sigma_j$ (known as the anti-ferromagnetic Ising model).

The Ising model can be used to study a model membrane by treating the lipids in one leaflet of the bilayer as fixed on a triangular lattice. Instead of interpreting the spins of the Ising model as being up or down we can allow each site to take on an arbitrary number of discrete states and interpret them as being different lipid species. The number of spins in each state is held fixed to conserve the mole fractions of each lipid species. In this way of defining the Ising model we must change the coupling constant depending on what lipids are neighboring each other,

$$\mathcal{H} = \sum_{\langle i,j \rangle} -J(\sigma_i, \sigma_j). \tag{1.4}$$

Where $J(\cdot, \cdot)$ is now a symmetric function that maps the pair-wise interaction energies of lipid species. The sign of J tells us about which lipid species attract or repel others, $J(A, B) > 0$ implies that lipid A is attracted to lipid B while $J(A, B) < 0$ implies that they repel. When just two lipid species are used, this model can be reduced down to a simple Ising model and exhibits exactly the same behavior. If three or more lipids are used, the energy landscape becomes highly non-trivial and the lipids will arrange themselves to minimize frustration. This way of modeling lipid membranes has been successful in capturing the tendency of lipids to phase separate [71, 52] and the thermodynamic behavior of the constituent lipids [48, 51,

53]. When appropriate update routines are used, the Ising model can also be used to study the diffusion dynamics of membranes [39].

Lattice Models

One weakness of the Ising model approach to lipid membranes is the hard constraint that each lipid (or coarse grained lipid analog) must be fixed on a planar sheet. For real membrane systems one key component of the phase sorting and energetic behavior lies in the three dimensional conformation of the membrane in space. This involves the curvature energies of the membrane and can be described using the Helfrich energy functional [46, 47, 15, 58],

$$\mathcal{H}_{\text{curve}} = \gamma L + \iint_S [\kappa H^2 + \bar{\kappa} G] dA \tag{1.5}$$

where S is the three dimensional surface describing the membrane shape, H and G are the **mean curvature** and **Gaussian curvature**, and κ and $\bar{\kappa}$ are the bending modulus and saddle-splay modulus [74]. When such contributions to the energy landscape cannot be ignored it is necessary to use a lattice model. Our lattice model approximates the membrane shape using a triangulated lattice [127], which can approximate any three dimensional surface to arbitrary precision.

Each vertex of the lattice is assigned an order parameter, ϕ, which represents properties of the lipid membrane, such as local phase or composition. The vertices interact with their nearest neighbors through an Ising like Hamiltonian to produce very similar behavior to an Ising model. The coupling to the shape occurs by presuming that the bending/saddle-splay moduli depend on ϕ. This coupling of shape and local composition creates a frustration in the energy functional that can lead to non-trivial morphologies [8, 54, 59, 70]. Lattice models are useful because they capture membrane shape, which is known to be important to the native behavior of model and cell membranes [80, 117, 108].

The lattice model has been used to successfully simulate many of the curvature-related phenomena of the membrane. Lattice models were able to study the kinetics of phase separation close to a critical point and showed that allowing the membrane to deform substantially slowed this process [113, 114]. The irregular shapes of organelles were modeled by including an anisotropic elasticity (a vector field related to molecular orientation) [87]. Curvature on small vesicles was shown to cause buckling into polyhedral shapes, which helped explain the geometric shapes of virus capsids [106]. It was also shown to stabilize multiple membrane domains [49]. This success in modeling equilibrium phase morphologies that had multiple distinct phase domains motivated the use of a lattice model in our exploration of curvatures role in modulated phases. It was found that appropriate parameters produced equilibrium modulated phases on the surface of GUVs [5, 36].

In practice, lattice models are very coarse grained. In each of the cited studies, the vertices of the lattice represent multiple lipids acting as a single degree of freedom. Usually the vertices are treated in a binary manner, as either L_d or L_o, to represent a two-phase membrane. The very same coarse grained approach that makes lattice models so versatile can also lead to rather tricky complications. When degrees of freedom are lost, the energetic parameters (such as the Ising coupling J [26], the line tension γ [4], or bending modulus [60]) that govern microscopic behavior cease to be meaningful. It is necessary to keep this in mind when interpreting results of coarse grained simulations. In some cases renormalization group techniques can be used to determine how these quantities change under graining and avoid this subtle problem.

Bibliography

[1] K. Akashi, H. Miyata, H. Itoh, and K. Kinosita. Preparation of giant liposomes in physiological conditions and their characaterization under and optical microscope. *Bio-*

physical Journal, 71:3242–3250, 1996.

[2] S. Akimov, P. Kuzmin, J. Zimmerberg, and F. Cohen. Lateral tension increases the lines tension between two domains in a lipid bilayer membrane. *Phys. Rev. E.*, 75: 011919, 2007.

[3] P. Almeida and W. Vaz. *Handbook of Biological Physics*, volume 1. Elsevier Science B.V., 1995.

[4] J. Amazon and G. Feigenson. Pending. *Phys. Rev. E.*

[5] J. Amazon, S. Goh, and G. Feigenson. Competition between line tension and curvature stabilizes modulated phase patterns on the surface of giant unilamellar vesicles: a simulation study. *Phys. Rev. E.*, 87:022708, 2013.

[6] R. Anderson and K. Jacobson. A role for lipid shells in targeting proteins to caveole, rafts, and other lipid domains. *Science*, 296:1821–1825, 2002.

[7] M. Angelova and D. Dimitrov. Liposome electroformation. *Farad. Discuss.*, 81:303–311, 1986.

[8] G. Arreaga, R. Capovilla, C. Chryssomalakos, and J. Guven. Area-constrained planar elastica. *Phys. Rev. E.*, 65:031801, 2002.

[9] A. Ayuyan and F. Cohen. Lipid peroxides promote large rafts: Effects of excitation of probes in fluorescence microscopy and electrochemical reactions during vesicle formation. *Biophysical Journal*, 91:2172–2183, 2006.

[10] A. Ayuyan and F. Cohen. Raft composition at physiological temperature and ph in the absence of detergents. *Biophysical Journal*, 94:2654–2666, 2008.

[11] T. Baumgart, S. Hess, and W. Webb. Imaging coexisting fluid domains in biomembrane models coupling curvature and line tension. *Nature*, 425:821–824, 2003.

[12] D. Bennett and P. Tieleman. Free energies of lipid-lipid interactions in membranes. *Annu. Rep. Comput. Chem.*, 5:3–17, 2009.

[13] D. Brown. Lipid rafts, detergent-resistant membranes, and raft targeting signals. *Physiol.*, 21:430–439, 2006.

[14] D. Brown and J. Rose. Sorting of gpi-anchored proteins to glycolipid-enriched membrane subdomains during transport to the apical cell surface. *Cell*, 68:533–544, 1992.

[15] F. Brown. Elastic modeling of biomembranes and lipid bilayers. *Annu. Rev. Phys. Chem.*, 59:685–712, 2008.

[16] J. Buboltz. Steady-state probe-partitioning fluorescence resonance energy transfer: A simple and robust tool for the study of membrane phase behavior. *Phys. Rev. E.*, 76: 021903, 2007.

[17] J. Buboltz. A more efficient device for preparing model-membrane liposomes by the rapid solvent exchange method. *Rev. Sci. Instrum.*, 80:124301, 2009.

[18] J. Buboltz and G. Feigenson. A novel strategy for the preparation of liposomes: Rapid solvent exchange. *Biochim. Biophys. Acta.*, 1417:232–245, 1999.

[19] J. Buboltz, C. Bwalya, S. Reyes, and D. Kamburov. Stern-volmer modeling of steady-state forster energy transfer between dilute, freely diffusing membrane-bound fluorophores. *J. Chem. Phys.*, 127:215101, 2007.

[20] H. Cheng, Megha, and E. London. Preparation and properties of asymmetric vesicles that mimic cell membranes. *J. Biol. Chem.*, 284:6079–6092, 2009.

[21] J. Dai, M. Alwarawrah, and J. Huang. Instability of cholesterol clusters in lipid bilayers and the cholesterol's umbrella effect. *J. Phys. Chem. B*, 114:840–848, 2010.

[22] S. Davis, P. Kumar, M. Sperotto, and M. Laradji. Predictions of phase separation in three-component lipid membranes by the martini force field. *J. Phys. Chem. B*, 117: 4072–4080, 2013.

[23] D. Dean and M. Manghi. Fluctuation-induced interactions between domains in membranes. *Phys. Rev. E.*, 74:021916, 2006.

[24] A. Demchenko and S. Yesylevskyy. Nanoscopic description of biomembrane electrostatics: Results of molecular dynamics simulations and fluorescence probing. *Chem. Phys. Lipids*, 160:63–84, 2009.

[25] J. Dinic, P. Ashrafzadeh, and I. Parmryd. Actin filaments attachment at the plasma membrane in live cells cause the formation of ordered lipid domains. *Biochim. Biophys. Acta.*, 1828:1102–1111, 2013.

[26] S. Duane, R. Horgan, and D. Scott. Block spin techniques applied to the ising model. *Z. Phys. C*, 16:355–359, 1983.

[27] M. Edidin. Lipids on the frontier: A century of cell-membrane bilayers. *Nature Rev. Mol. Cell Biol.*, 5:414–418, 2003.

[28] E. Elson, E. Fried, J. Dolbow, and G. Genin. Phase separation in biological membranes: Integration of theory and experiment. *Annu. Rev. Biophys.*, 39:207–226, 2010.

[29] C. Esposito, A. Tian, S. Melamed, C. Johnson, S. Tee, and T. Baumgart. Flicker spectroscopy of thermal lipid bilayer domain boundary fluctuations. *Biophysical Journal*, 93:3169–3181, 2007.

[30] T. Estep and T. Thompson. Energy transfer in lipid bilayers. *Biophysical Journal*, 26: 195–208, 1979.

[31] G. Feigenson. Phase behavior of lipid mixtures. *Nat. Chem. Biol.*, 2:560–563, 2006.

[32] G. Feigenson. Phase boundaries and biological membranes. *Annu. Rev. Biophys. Biomol. Struct.*, 36:63–77, 2007.

[33] G. Feigenson and J. Buboltz. Ternary phase diagram of dipalmitoyl-pc/dilauroyl-pc/cholesterol: Nanoscopic domain formation driven by cholesterol. *Biophysical Journal*, 80:2775–2788, 2001.

[34] K. Field, D. Holowka, and B. Baird. Compartmentalized activation of the high affinity immunoglobulin e receptor within membrane domains. *J. Biol. Chem.*, 272:4276–4280, 1997.

[35] B. Fung and L. Stryer. Surface density determination in membranes by fluorescence energy transfer. *Biochemistry*, 17:5241, 1978.

[36] S. Goh, J. Amazon, and G. Feigenson. Toward a better raft model: Modulated phases in the four-component bilayer, dspc/dopc/popc/chol. *Biophysical Journal*, 104:853–862, 2013.

[37] D. Goswami, K. Gowrishankar, S. Bilgrami, S. Ghosh, R. Raghupathy, R. Chadda, R. Vishwakarma, M. Rao, and S. Mayor. Nanoclusters of gpi-anchored proteins are formed by cortical actin-driven activity. *Cell*, 135:1085–1097, 2008.

[38] J. Groves. Bending mechanics and molecular organization in biological membranes. *Annu. Rev. Phys. Chem.*, 58:697–717, 2007.

[39] A. Hac, H. Seeger, M. Fidorra, and T. Heimburg. Diffusion in two-component lipid membranes-a fluorescence correlation spectroscopy and monte carlo simulation study. *Biophysical Journal*, 88:317–333, 2005.

[40] A. Hammond, F. Heberle, T. Baumgart, D. Holowka, B. Baird, and G. Feigenson. Crosslinking a lipid raft component triggers liquid ordered-liquid disordered phase separation in model plasma membranes. *PNAS*, 102:6320–6325, 2005.

[41] J. Hancock. Lipid rafts: Conentious only from simplistic standpoints. *Nature Rev. Mol. Cell Biol.*, 7:456–462, 2006.

[42] T. Harder, P. Scheiffele, P. Verkade, and K. Simons. Lipid domain structure of the plasma membrane revealed by patching of membrane components. *J. Cell Bio.*, 141: 929–942, 1998.

[43] F. Heberle, J. Wu, S. Goh, R. Petruzielo, and G. Feigenson. Comparison of three ternary lipid bilayer mixtures: Fret and esr reveal nanodomains. *Biophysical Journal*, 99:3309–3318, 2010.

[44] F. Heberle, M. Doktorova, S. Goh, R. Standaert, J. Katsaras, and G. Feigenson. Hybrid and nonhybrid lipids exert common effects on membrane raft size and morphology. *J. Am. Chem. Soc.*, 135:14932–14935, 2013.

[45] F. Heberle, R. Petruzielo, J. Pan, P. Drazba, N. Kucerka, R. Standaert, G. Feigenson, and J. Katsaras. Bilayer thickness mismatch controls domain size in model membranes. *J. Am. Chem. Soc.*, 135:6853–6859, 2013.

[46] W. Helfrich and R. Servuss. Undulations, steric interaction and cohesion of fluid membranes. *Il Nuovo Cimento*, 3:137–151, 1984.

[47] J. Henriksen and J. Ipsen. Thermal undulations of quasi-spherical vesicles stabilized by gravity. *Eur. Phys. J. E*, 9:365–374, 2002.

[48] A. Honerkamp-Smith, P. Cicuta, M. Collins, S. Veatch, M. Nijs, M. Schick, and S. Keller. Line tensions, correlation lengths, and critical exponents in lipid membranes near critical points. *Biophysical Journal*, 95:236–246, 2008.

[49] J. Hu, T. Weikl, and R. Lipowsky. Vesicles with multiple membrane domains. *Soft Matter*, 7:6092–6102, 2011.

[50] M. Hu, J. Briguglio, and M. Deserno. Determining the gaussian curvature modulus of lipid membranes in simulations. *Biophysical Journal*, 102:1403–1410, 2012.

[51] J. Huang. Exploration of molecular interactions in cholesterol superlattices: Effect of multibody interactions. *Biophysical Journal*, 83:1014–1025, 2002.

[52] J. Huang and G. Feigenson. Monte carlo simulation of lipid mixtures: Finding phase separation. *Biophysical Journal*, 65:1788–1794, 1993.

[53] J. Huang and G. Feigenson. A microscopic interaction model of maximum solubility of cholesterol in lipid bilayers. *Biophysical Journal*, 76:2142–2157, 1999.

[54] T. Idema and C. Storm. Analytical expressions for the shape of axisymmetric membranes with multiple domains. *Eur. Phys. J. E*, 34:67, 2011.

[55] E. Ikonen and K. Simons. Protein and lipid sorting from the trans-golgi network to the plasma membrane in polarized cells. *Sem. Cell Devel. Biol.*, 9:503–509, 1998.

[56] A. Imperio and L. Reatto. Microphase separation in two-dimensional systems with competing interactions. *J. Chem. Phys.*, 124:164712, 2006.

[57] Y. Kaizuka and J. Groves. Bending-mediated superstructural organizations in phase-separated lipid membranes. *New J. of Phys.*, 12:095001, 2010.

[58] T. Kawakatsu, D. Andelman, K. Kawasaki, and T. Taniguchi. Phase transitions and shapes of two component membranes and vesicles i: Strong segregation limit. *J. Phys. II France*, 3:971–997, 1993.

[59] K. Khairy and J. Howard. Minimum-energy vesicle and cell shapes calculated using spherical harmonics parameterization. *Soft Matter*, 7:2138–2143, 2011.

[60] H. Kleinert. Thermal softening of curvature elasticity in membranes. *Phys. Lett.*, 114A:263–268, 1986.

[61] T. Konyakhina, S. Goh, J. Amazon, F. Heberle, J. Wu, and G. Feigenson. Control of nanoscopic-to-macroscopic transition: Modulated phases in 4-component dspc/dopc/popc/chol giant unilamellar vesicles. *Biophysical Journal*, 101:L8–L10, 2011.

[62] T. Konyakhina, J. Wu, J. Mastroianni, F. Heberle, and G. Feigenson. Phase diagram of a 4-component lipid mixture: Dspc/dopc/popc/chol. *Biochim. Biophys. Acta.*, 1828: 2204–2214, 2013.

[63] J. Korlach, P. Schwille, W. Webb, and G. Feigenson. Characterization of lipid bilayer phases by confocal microscopy and fluorescence correlation spectroscopy. *PNAS*, 96: 8461–8466, 1999.

[64] E. Landau and J. Rosenbusch. Lipidic cubic phases: A novel concept for the crystallization of membrane proteins. *PNAS*, 93:14532–14535, 1996.

[65] I. Levental, F. Byfield, P. Chowdhury, F. Gai, T. Baumgart, and P. Janmey.

Cholesterol-dependant phase separation in cell-derived giant plasma-membrane vesicles. *Biochem. J.*, 424:163–167, 2009.

[66] I. Levental, D. Lingwood, M. Gryzbek, U. Coskun, and K. Simons. Palmitoylation regulates raft affinity for the majority of integral raft proteins. *PNAS*, 107:22050–22054, 2010.

[67] I. Levental, M. Grzybek, and K. Simons. Raft domains of variable properties and compositions in plasma membrane vesicles. *PNAS*, 108:11411–11416, 2011.

[68] D. Lichtenberg, F. Goni, and H. Heerklotz. Detergent-resistant membranes should not be identified with membrane rafts. *J. tibs.*, 30:430–436, 2005.

[69] D. Lingwood and K. Simons. Lipid rafts as a membrane organizing principle. *Science*, 327:46–50, 2010.

[70] R. Lipowsky. The conformation of membranes. *Nature*, 349:475–481, 1991.

[71] B. Machta, S. Papanikolaou, J. Sethna, and S. Veatch. Minimal model of plasma membrane heterogeneity requires coupling cortical actin to criticality. *Biophysical Journal*, 100:1668–1677, 2011.

[72] S. Marrink and A. Mark. Molecular dynamics simulation of formation, structure, and dynamics of small phospholipid vesicles. *J. Am. Chem. Soc.*, 125:15233–15242, 2003.

[73] S. Marrink, J. Risselada, and A. Mark. Simulation of gel phase formation and melting in lipid bilayers using a coarse grained model. *Chem. Phys. Lipids*, 135:223–244, 2005.

[74] D. Marsh. Elastic curvature constants of lipid monolayers and bilayers. *Chem. Phys. Lipids*, 144:146–159, 2006.

[75] N. Metropolis, A. Rosenbluth, M. Rosenbluth, and A. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092, 1953.

[76] S. Munroe. Lipid rafts: Elusive or illusive? *Cell*, 115:377–388, 2003.

[77] D. Nicolau, J. Hancock, and K. Burrage. Sources of anomalous diffusion on cell membranes: A monte carlo study. *Biophysical Journal*, 92:1975–1987, 2007.

[78] H. Nymeyer and H. Zhou. A method to determine dielectric constants in nonhomogeneous systems: Application to biological membranes. *Biophysical Journal*, 94: 1185–1193, 2008.

[79] B. Palmieri and S. Safran. Hybrid lipids increase nanoscale fluctuation lifetimes in mixed membranes. *Phys. Rev. E.*, 88:032708, 2013.

[80] I. Parmryd and Bjorn Onfelt. Consequences of membrane topography. *febs.*, 280: 2775–2784, 2013.

[81] P. Pathak and E. London. Measurement of lipid nanodomain (raft) formation and size in sphingomyelin/popc/cholesterol vesicles shows tx-100 and transmembrane helices increase domain size by coalescing preexisting nanodomains but do not induce domain formation. *Biophysical Journal*, 101:2417–2425, 2011.

[82] J. Pecreaux, H. Dobereiner, J. Prost, J. Joanny, and P. Bassereau. Refined contour analysis of giant unilamellar vesicles. *Eur. Phys. J. E*, 13:277–290, 2004.

[83] R. Petruzielo, F. Heberle, P. Drazba, J. Katsaras, and G. Feigenson. Phase behavior and domain size in sphingomyelin-containing lipid bilayers. *Biochim. Biophys. Acta.*, 1828:1302–1313, 2013.

[84] L. Pike. Lipid rafts: Bringing order to chaos. *J. Lipid Res.*, 44:655–667, 2003.

[85] E. Pohl. *Advances in Planar Lipid Bilayers and Liposomes*, volume 1. Elsevier Inc., 2005.

[86] T. Politano, V. Froude, B. Jing, and Y. Zhu. Ac-electric field dependent electroformation of giant lipid vesicles. *Colloids Surf., B*, 79:75–82, 2010.

[87] N. Ramakrishnan, P. Sunil Kumar, and J. Ipsen. Modeling anisotropic elasticity of fluid membranes. *Macromolecul. Theory Simul.*, 20:446–450, 2011.

[88] A. Raudino and D. Mauzerall. Dielectric properties of the polar head group region of zwitterionic lipid bilayers. *Biophysical Journal*, 50:441–449, 1986.

[89] S. Reed. In c on it's own terms: A statistical and historical view. *Perspectives of New Music*, 49:47–78, 2011.

[90] B. Reynwar, G. Illya, V. Harmandaris, M. Muller, K. Kremer, and M. DeSerno. Aggregation and vesiculation of membrane proteins by curvature-mediated interactions. *Nature*, 447:461–464, 2007.

[91] S. Rozovsky, Y. Kaizuka, and J. Groves. Formation and spatio-temporal evolution of periodic structures in lipid bilayers. *J. Am. Chem. Soc.*, 127:36–37, 2005.

[92] L. Schafer and S. Marrink. Partitioning of lipids at domain boundaries in model membranes. *Biophysical Journal*, 99:L91–L93, 2010.

[93] L. Schermelleh, R. Heintzmann, and H. Leonhardt. A guide to super-resolution fluorescence microscopy. *J. Cell Bio.*, 190:165–175, 2010.

[94] M. Schick. Membrane heterogeneity: Manifestation of a curvature-induces microemulsion. *Phys. Rev. E.*, 85:031902, 2011.

[95] A. Seddon, P. Curnow, and P. Booth. Membrane proteins, lipids and detergents: Not just a soap opera. *Biochim. Biophys. Acta.*, 1666:105–117, 2004.

[96] J. Seddon. Structure of the inverted hexagonal (hii) phase, and non-lamellar phase transitions of lipids. *Biochim. Biophys. Acta.*, 1031:1–69, 1990.

[97] S. Semrau, T. Idema, L. Holtzer, T. Schmidt, and C. Storm. Accurate determination of elastic parameters for multicomponent membranes. *Phys. Rev. Lett.*, 100:088101, 2008.

[98] S. Semrau, T. Idema, T. Schmidt, and C. Storm. Membrane-mediated interactions measured using membrane domains. *Biophysical Journal*, 96:4906–4915, 2009.

[99] P. Sengupta, B. Baird, and D. Holowka. Lipid rafts, fluid/fluid phase separation, and their relevance to plasma membrane structure and function. *Sem. Cell Devel. Biol.*, 18:583–590, 2007.

[100] M. Seul and D. Andelman. Domain shapes and patterns: The phenomenology of modulated phases. *Science*, 267:476–483, 1995.

[101] K. Simons and E. Ikonen. Functional rafts in cell membranes. *Nature*, 387:569–572, 1997.

[102] K. Simons and G. Van Meer. Lipid sorting in epithelial cells. *Biochemistry*, 27:6197–6202, 1988.

[103] K. Simons and D. Toomre. Lipid rafts and signal transduction. *Nature Rev. Mol. Cell Biol.*, 1:31–41, 2000.

[104] K. Simons and W. Vaz. Model systems, lipid rafts, and cell membranes. *Annu. Rev. Biophys. Biomol. Struct.*, 33:269–295, 2004.

[105] S. Singer and G. Nocolson. The fluid mosaic model of the structure of cell membranes. *Science*, 175:720–731, 1972.

[106] R. Sknepnek, G. Vernizzi, and M. Olvera de la Cruz. Buckling of multicomponent elastic shells with line tension. *Soft Matter*, 8:636–644, 2011.

[107] A. Smondyrev and M. Berkowitz. Molecular dynamics simulation of dipalmitoylphosphatidylcholine membrane with cholesterol sulfate. *Biophysical Journal*, 78:1672–1680, 2000.

[108] S. Sonnino, A. Prinetti, L. Mauri, V. Chigorno, and G. Tettamanti. Dynamic and structural properties of sphingolipids as driving forces for the formation of membrane domains. *Chem. Rev.*, 106:2111–2125, 2006.

[109] M. Sperotto, S. May, and A. Baumgartner. Modelling of proteins in membranes. *Chem. Phys. Lipids*, 141:2–29, 2006.

[110] J. Steim, M. Tourtellotte, J. Reinert, R. McElhaney, and R. Rader. Calorimetric evidence for the liquid-crystalline state of lipids in a biomembrane. *Biochemistry*, 63: 104–109, 1969.

[111] M. Swamy, L. Ciani, M. Ge, A. Smith, D. Holowka, B. Baird, and J. Freed. Coexisting domains in the plasma membranes of live cells characterized by spin-label esr spectroscopy. *Biophysical Journal*, 90:4452–4465, 2006.

[112] Y. Tamba, H. Terashima, and M. Yamazaki. A membrane filtering method for the purification of giant unilamellar vesicles. *Chem. Phys. Lipids*, 164:351–358, 2011.

[113] T. Taniguchi. Shape deformations and phase separation dynamics of two-component vesicles. *Phys. Rev. Lett.*, 76:4444–4447, 1996.

[114] T. Taniguchi, M. Yanagisawa, and M. Imai. Numerical investigations of the dynamics of two-component vesicles. *J. Phys. Condens. Matter*, 23:284103, 2011.

[115] A. Tian, C. Johnson, W. Wang, and T. Baumgart. Line tension at fluid membrane domain boundaries measured by micropipette aspiration. *Phys. Rev. Lett.*, 98:208102, 2007.

[116] A. Travesset. Effect of dipolar moments in domain sizes of lipid bilayers and monolayers. *J. Chem. Phys.*, 125:084905, 2006.

[117] T. Ursell, W. Klug, and R. Phillips. Morphology and interaction between lipid domains. *PNAS*, 106:13301–13306, 2009.

[118] S. Veatch, O. Soubias, S. Keller, and K. Gawrisch. Critical fluctuations in domain-forming lipid mixtures. *PNAS*, 104:17650–17655, 2007.

[119] M. Vist and J. Davis. Phase equilibria of cholesterol/dipalmitoylphosphatidylcholine mixtures: ^{2}h nuclear magnetic resonance and differential scanning calorimetry. *Biochemistry*, 29:451–464, 1990.

[120] L. Wang. Measurements and implications of the membrane dipole potential. *Annu. Rev. Biochem.*, 81:615–635, 2012.

[121] L. Wang, P. Bose, and F. Sigworth. Using cryo-em to measure the dipole potential of a lipid membrane. *PNAS*, 103:18528–18533, 2006.

[122] T. Wang and J. Silvius. Cholesterol does not induce segregation of liquid-ordered domains in bilayers modeling the inner leaflet of the plasma membrane. *Biophysical Journal*, 81:2762–2773, 2001.

[123] P. Wu and L. Brand. Resonance energy transfer: Methods and applications. *Anal. Biochem.*, 218:1–13, 1994.

[124] Y. Yoon, J. Hale, P. Petrov, and P. Cicuta. Mechanical properties of ternary lipid membranes near a liquid-liquid phase separation boundary. *J. Phys. Condens. Matter*, 22:062101, 2010.

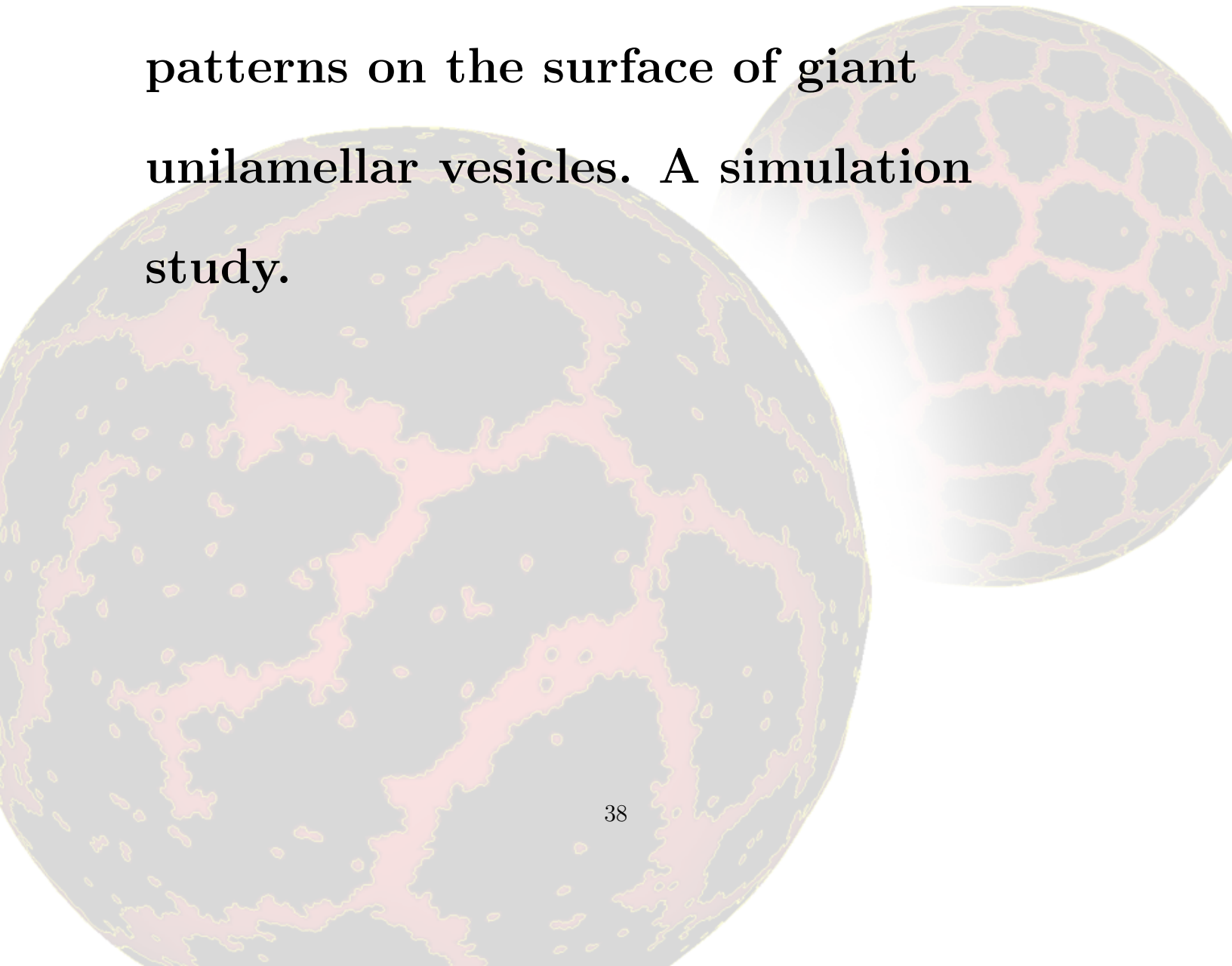
[125] J. Zhao, J. Wu, F. Heberle, T. Mills, P. Klawitter, G. Huang, G. Costanza, and G. Feigenson. Phase studies of model biomembranes: Complex behavior of dspc/dopc/cholesterol. *Biochim. Biophys. Acta.*, 1768:2764–2776, 2007.

[126] J. Zhao, J. Wu, H. Shao, F. Kong, N. Jain, G. Hunt, and G. Feigenson. Phase studies of model biomembranes: Macroscopic coexistence of lα+lβ, with light-induced coexistence of lα+lo phases. *Biochim. Biophys. Acta.*, 1768:2777–2786, 2007.

[127] D. Zorin. Curvature-based energy for simulation and variational modeling. *Shape Modeling and Applications, 2005 International Conference*, 1:196–204, 2005.

Chapter 2

Competition between line tension and curvature stabilizes modulated phase patterns on the surface of giant unilamellar vesicles. A simulation study.



2.1 Abstract

When prepared in the liquid-liquid coexistence region, the four component lipid system distearoyl-phosphatidylcholine (DSPC)/dioleoyl-phosphatidylcholine (DOPC)/palmitoyl,oleoyl-phosphatidylcholine (POPC)/Cholesterol with certain ratios of DOPC and POPC shows striking modulated phase patterns on the surface of giant unilamellar vesicles (GUVs). In this simulation study we show that the morphology of these patterns can be explained by the competition of line tension (which tends to favor large round domains) and curvature, as specified by the Helfrich energy functional. In this study we use a Monte-Carlo simulation on the surface of a GUV to determine the equilibrium shape and phase morphology. We find that the patterns arising from these competing interactions very closely approximate those observed, the patterned morphologies represent thermodynamically stable configurations, and that the geometric nature of these patterns is closely tied to the relative and absolute values of the model parameters.

2.2 Introduction

Model systems provide an important way to study and understand the behavior of multicomponent lipid bilayer membranes. The three component lipid system, distearoyl-phosphatidylcholine (DSPC)/dioleoyl-phosphatidylcholine (DOPC)/cholesterol (chol) has a well characterized phase diagram, with a region of liquid-liquid (L_o + L_d) coexistence that is readily observable in giant unilamellar vesicle (GUV) studies as large round domains [29, 27]. This mixture is a useful model for understanding the general nature of bilayers containing a high melting lipid (DSPC), a low melting lipid (DOPC), and cholesterol.

If we replace the low melting lipid, DOPC, with POPC (palmitoyl, oleoyl-phosphatidylcholine), we find a stark difference in the phase morphology. Performing GUV imaging studies on

the POPC-containing system reveals that the liquid-liquid coexistence region appears uniform, unlike the macroscopic phase domains seen with DOPC-containing mixtures. However, FRET, ESR [8], and neutron scattering studies [9] show that liquid-liquid coexistence is present with POPC. This observation implies that phase separation occurs in the POPC-containing system on the nanometer scale, thus not resolvable by ordinary light microscopy. Given the large body of data from studies of animal cell plasma membranes that supports the occurrence of $L_o + L_d$ phase domains [24, 22, 16], and given that the size scale, shapes, and connectivities of phase-separated domains might be involved in the fundamental behaviors of animal cells, understanding the membrane in terms of a nonrandom physical mixture might be important.

A four-component mixture containing DSPC, DOPC, POPC, and cholesterol enables study of the full range of phase mophologies, from domain size scale of nanometers to microns, that could occur in cell membranes. On the DOPC-heavy side of this phase diagram we expect large macroscopic domains, whereas on the POPC-heavy side we expect GUVs to appear uniform (nanodomains). The relative amounts of DOPC and POPC can be controlled precisely in order to study the macro-to-nano transition [15]. The relative amount of DOPC and POPC can be described by ρ, given as,

$$\rho \equiv \frac{\chi_{DOPC}}{\chi_{DOPC} + \chi_{POPC}}, \tag{2.1}$$

where χ_{DOPC} and χ_{POPC} are the mole fractions of DOPC and POPC respectively. In this way, $\rho = 1.00$ corresponds to a three-component mixture with DOPC as the only low-melting component and $\rho = 0.00$ corresponds to a three-component mixture with POPC as the only low-melting component. The trajectory defined by varying ρ from 0.00 to 1.00 pierces the tetrahedral composition space, with endpoints located on the POPC and DOPC faces. For both experimental and simulation studies, the mole fractions of DSPC and CHOL are held

fixed unless otherwise stated.

When DSPC and CHOL are held at particular mole fractions within the liquid-liquid coexistence region, for example $\chi_{DSPC} = 0.45$ and $\chi_{CHOL} = 0.25$, and the value of ρ is varied, a patterning of the phases is observed in the composition range, $0.15 < \rho < 0.25$ [15]. This range can be termed the "modulated phase window" and for these particular compositions is characterized by thin stripes of L_d phase within a more abundant L_o phase, as shown in figure 2.1. The overall pattern can resemble striped or honeycomb-like structures, each with a characteristic periodicity. For this particular composition ($\chi_{DSPC} = 0.45, \chi_{CHOL} = 0.25$), macroscopic domains are observed when $\rho > 0.25$ and uniform GUVs are observed when $\rho < 0.15$ [15].

The underlying mechanism of the formation of modulated phases in this system has not previously been understood. Our aim is to model and simulate the formation of these modulated phases to better understand the transition taking place as mixture composition moves through this ρ window. We can then use this model to make predictions about the nature of nanodomains present in the pure POPC system and eventually the nature of phase separation on the plasma membrane.

2.3 Materials and Methods

2.3.1 GUV preparation and microscopy

GUV samples were prepared as described in [15] with the following modifications: GUVs were swelled at 55^{o}C in either 100mM sucrose or 100mM glucose, then cooled to room temperature (23^{o}C) over 12 hours. Samples were harvested into microcentrifuge tubes (Fisher Scientific) using large orifice pipet tips (Fisher Scientific), and let settle for about 2 hours before observation.

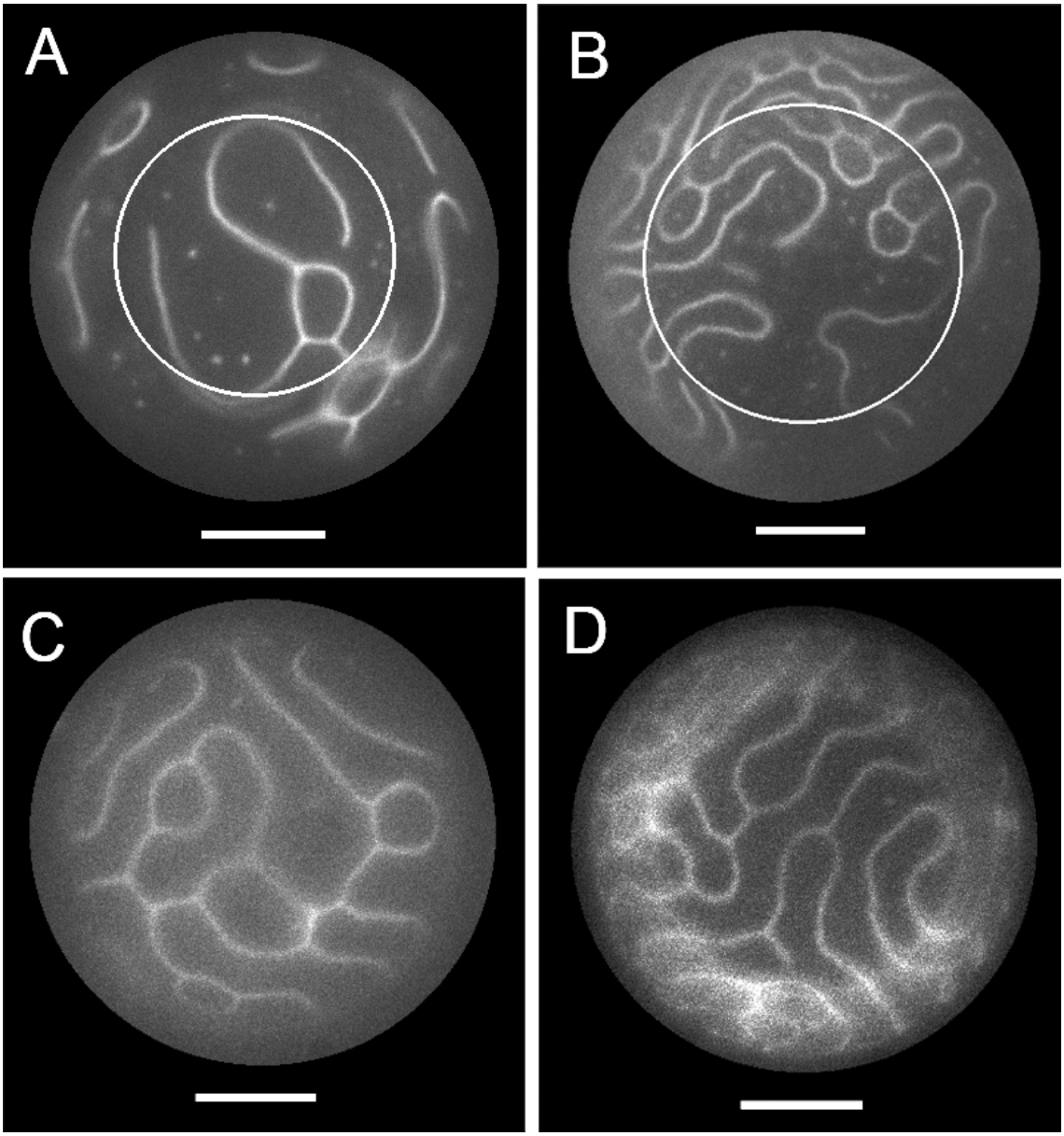


Figure 2.1: Four examples of modulated phase patterns observed using wide field fluorescence microscopy. Images are cropped and contrast adjusted. (A) and (B), composite images of focused slices of the GUV, the white ring marking the boundary of the composite image. GUV compositions DSPC/DOPC/POPC/Chol: (A) 0.487/0.0625/0.188/0.263, (B) 0.45/0.075/0.225/0.25, (C) and (D) 0.395/0.12/0.18/0.305. Dye C12:0 DiI (0.02 mol%) partitions into L_d. Scale bars, 10μm, temperature 23^{o}C.

Widefield microscopy was performed on a Nikon Diaphot-TMD inverted microscope at 23^oC using a 60X 1.4NA oil immersion objective. To minimize light-induced artifacts, GUVs were first located in bright field mode before illumination for fluorescence. Samples contained 0.02 mol% C12:0 DiI, imaged with 535-550 nm excitation and 565-610 nm emission. Images were collected with a Photometrics charge-coupled device camera CoolSNAPHQ2 (Tucson, Arizona).

Phospholipids were purchased from Avanti Polar Lipids (Alabaster, AL), cholesterol from Nu Chek Prep (Elysian, MN) and the fluorescent dye C12:0-DiI (1,1'-didodecyl-3,3,3',3'-tetramethylindocarbocyanine perchlorate) from Invitrogen (Carlsbad, CA). Concentrations of phospholipid stocks were determined to $< 1\%$ error with inorganic phosphate assay [14], and purity checked with thin layer chromatography in chloroform/methanol/water solvent. Cholesterol at defined concentration was prepared by standard gravimetric procedures. Fluorescent dye concentrations were determined using absorption spectroscopy on an HP 8452A spectrophotometer (Hewlett-Packard, Palo Alto, CA).

2.3.2 Simulation model

To approach the problem of explaining the appearance of modulated phases, we constrain our model of the observed patterns in two ways:

1. Modulated phases are at a state of thermodynamic equilibrium. This is supported by the observation that the patterns do not change over the observation times, persisting for hours and even days.

2. Bilayers are phase-separated, the two coexisting phases are fluid, and the value of ρ changes only the material parameters describing the energetics of that phase. These assumptions are reasonable because the compositions under consideration are squarely within

the liquid-liquid coexistence region of both of the $\rho = 0.00$ and $\rho = 1.00$ faces of the phase diagram tetrahedron [8]. This simplifies the mathematical model, as we can drop terms in the energy functional that are not related to morphology (such as any term that depends only on the local composition).

Our approach uses a competing interactions model [23], which has been shown to produce modulated phases in many systems. The formalism states that multiple fields (order parameters) that couple in a way that opposes the formation of a single domain with minimal boundary can form equilibrium honeycomb and striped patterns.

The fields we consider are the local composition and the local curvature of the membrane [3, 10, 13]. These couple through the composition-dependent material properties [21, 18, 28, 19, 4], which dictate the energetics of bending and stretching the membrane. We implement this model in simulation to find the thermodynamic equilibrium state. To do so we use a Monte-Carlo simulation to sample the configuration space of fields, then we use the Metropolis Algorithm [10] to ensure that energy is minimized to within thermal noise.

Similar simulations have been performed, including the work of Lipowsky et al. that showed how curvature and line tension were able to stabilize as many as seven domains and give rise to a variety of GUV shapes [10]. Simulation studies of Fan et al. showed that more complicated models, such as lipid recycling, can stabilize non-equilibrium patterns on a flat membrane [6]. Other models for the stabilization of multiple/patterned domains have also been studied including a general competing interaction model [12] and the effects of dipolar repulsion between lipids [25, 17, 20]. It has been shown that electrostatics are too short-range to account for the many micron length scale we observe in modulated phases [15]

To begin modeling the modulated phases we first formulate an energy functional. The three fields that our energy functional depends on are the local phase, ϕ, the mean curvature, H, and the Gaussian curvature G.

2.3.3 Local phase field

The local phase at a point, r, on the membrane surface is given by the $\phi(r)$ field,

$$\phi(r) = \begin{cases} 0 & r \in L_d \\ 1 & r \in L_o \end{cases} \tag{2.2}$$

This binary field allows us to define the phase morphology on the surface. We also assume that the total amount of each phase is fixed (that is, we choose where we are along a given thermodynamic tieline). If we specify the fraction of surface in the L_o phase by P, then this constraint can be written,

$$\frac{1}{A_{tot}} \iint \phi(r) dA = P, \tag{2.3}$$

where A_{tot} is the total area of the membrane surface. The value of A_{tot} is also considered fixed because of the large elastic modulus of lipid bilayers [5, 19, 10].

2.3.4 Mean curvature field

The mean curvature can be defined as the divergence of the surface normals,

$$H(r) = \nabla \cdot \vec{n}(r), \tag{2.4}$$

which means that mean curvature can roughly be interpreted as the degree to which the normal vectors point towards ($H < 0$) or away from ($H > 0$) each other. For computational purposes we use an alternative definition of the mean curvature relating to the change in area under normal projection:

If each point in a small patch of area A is projected outwards a distance ΔR along the surface normals, then the change in area ΔA is related to the mean curvature by,

$$\Delta A = AH(r)\Delta R \tag{2.5}$$

which can be rearranged to give an explicit expression for $H(r)$.

$$H(r) = \frac{\Delta A}{A\Delta R} \tag{2.6}$$

This is known as the First Variation of Area formula, which lends itself easily to a general method for calculating curvature fields on a discrete surface. A diagram of this process is shown in Figure 2.4.

2.3.5 Gaussian curvature field

The Gaussian curvature can be locally defined by the Gauss-Bonnet theorem. Consider a small N-sided polygon with area A, centered on a point r, that is delimited by surface geodesics that meet at exterior angles θ_1, θ_2,...,θ_N. The Gaussian curvature at r is,

$$G(r) = \frac{1}{A}\left[2\pi - \sum_{i=1}^{N}\theta_i\right] \tag{2.7}$$

Gaussian curvature is roughly a measure of the extent to which parallel lines drawn on the surface are bent towards each other ($G > 0$) or away from each other ($G < 0$).

2.3.6 Energy functional

Since we assume that the membrane is already phase separated, our Hamiltonian has three major contributions [5, 10]: the line tension term, $\mathcal{H}_P$, the mean curvature term, $\mathcal{H}_H$, and the Gaussian curvature term, $\mathcal{H}_G$. Thus the Hamiltonian is expressed as,

$$\mathcal{H}[\phi, H, G] = \mathcal{H}_P + \mathcal{H}_H + \mathcal{H}_G \tag{2.8}$$

Line tension term ($\mathcal{H}_P$)

The line tension is an expression of the unfavorable energy required to make an interface between the two membrane phases [15, 7, 28]. It is defined by multiplying the total perimeter of the phase boundary, L, by a constant energy per unit length, γ,

$$\mathcal{H}_P = \gamma L \tag{2.9}$$

It can be shown that for a fixed amount of L_d and L_o phases, the minimal boundary is achieved when one large round domain is formed, maximizing the area-to-perimeter ratio. This drives the system towards macroscopic phase separation.

To determine the value of L strictly from the function $\phi(r)$ is the product of a complicated limiting process. In practice L is computed discretely and takes on the form of a simple sum.

Mean curvature term ($\mathcal{H}_H$)

The mean curvature and Gaussian curvature terms are both from the Helfrich formulation [5, 10]. This defines the energy of bending the membrane up to quadratic order in the mean and Gaussian curvatures. The mean curvature expression is given by,

$$\mathcal{H}_H = \iint \kappa(\phi(r))[H(r)]^2 dA, \tag{2.10}$$

where $\kappa(\phi)$ is the local bending modulus. This is defined as,

$$\kappa(\phi) = \begin{cases} \kappa_d & \phi = 0 \\ \kappa_o & \phi = 1 \end{cases} \tag{2.11}$$

where κ_d and κ_o are the bending modulus in the L_d and L_o phases respectively.

Gaussian curvature term ($\mathcal{H}_G$)

The Gaussian curvature term is also from the Helfrich functional and is defined as,

$$\mathcal{H}_G = \iint \bar{\kappa}(\phi(r))G(r)dA, \tag{2.12}$$

where $\bar{\kappa}(\phi)$ is defined similarly as,

$$\bar{\kappa}(\phi) = \begin{cases} \bar{\kappa}_d & \phi = 0 \\ \bar{\kappa}_o & \phi = 1 \end{cases} \tag{2.13}$$

where $\bar{\kappa}_d$ and $\bar{\kappa}_o$ are the Gaussian bending modulus in the L_d and L_o phases respectively.

Constraints

In addition to the three terms outlined above we also include two harmonic constraint terms to keep the area and volume of the GUV close to a fixed value (defined at the outset of the simulation).

$$\mathcal{H}_A = A_l(A_{tot} - A)^2 \tag{2.14}$$

$$\mathcal{H}_V = V_l(V_{tot} - V)^2 \tag{2.15}$$

where A_{tot} and V_{tot} are the fixed values of the area and volume respectively. The harmonic strengths A_l and V_l are in practice kept very large to allow very little fluctuation [10].

2.4 Discrete model

The formalism outlined above is the continuum description of a phase separated membrane's energetics. In order to simulate this system it is necessary to find a suitable discretization scheme that will allow us to faithfully mimic the conformations of a continuum membrane. To do this we use a triangulated lattice with an overall spherical topology [10].

This lattice can be described by a set of vertices, edges, and faces ($\{v_i\}$,$\{e_i\}$, and $\{f_i\}$) with N_v,N_e, and N_f elements respectively. The number of vertices, edges, and faces also must satisfy the Euler characteristic of a sphere, $N_v - N_e + N_f = 2$. To define the topology we must define the connectivity of the lattice. With n_i the number of neighbors that v_i [1] has (between 3 and 10), we define the three sets,

$$\{v_{ij}\} \quad \{e_{ij}\} \quad \{f_{ij}\}$$

where v_{ij} is the jth neighbor of vertex i. It is important to note that the sequence of vertices defined by,

$$v_{i0} \rightarrow v_{i1} \rightarrow ... \rightarrow v_{i(n_i-1)} \rightarrow v_{i0},$$

forms a counter clockwise loop around the parent vertex, v_i. The elements of $\{e_{ij}\}$ define the edges connecting the vertices v_i and v_{ij}. The elements of $\{f_{ij}\}$ define the faces containing the vertices v_i, v_{ij}, and $v_{i(j+1)}$. These elements are only defined for $0 \leq i < N_v$ and $0 \leq j < n_i$. The second index, j, is assumed to be modulo n_i unless stated otherwise. This geometric construction is shown in figure 2.2.

[1] The symbol v_i referes to the scalar index of the vertex, while the symbol $\vec{v}_i$ refers to the vector position of the vertex.

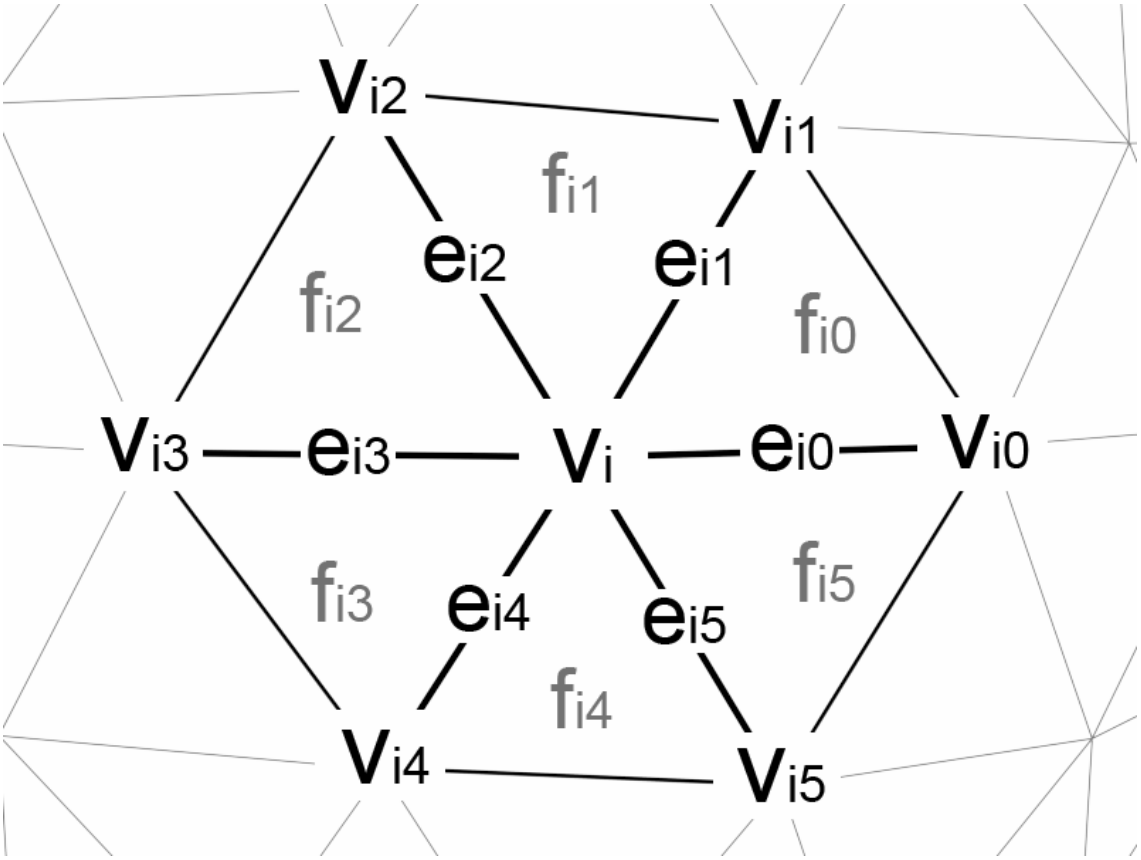


Figure 2.2: Geometric layout of a vertex and its neighboring vertices, edges, and faces. The sequence of neighbors form a counter clockwise loop allowing us to orient the surface normals outward.

2.4.1 Geometric properties

It is also useful to have the geometric parameters of the lattice at our disposal. Let f_i be a face delimited by the three vertices $\{\vec{w}_1, \vec{w}_2, \vec{w}_3\}$. The midpoint, normal, area, and volume associated with this face can all be defined from the geometry as,

$$\vec{m}(f_i) = \frac{1}{3}\left(\vec{w}_1 + \vec{w}_2 + \vec{w}_3\right) \tag{2.16}$$

$$\vec{n}(f_i) = \frac{(\vec{w}_2 - \vec{w}_1) \times (\vec{w}_3 - \vec{w}_1)}{\|(\vec{w}_2 - \vec{w}_1) \times (\vec{w}_3 - \vec{w}_1)\|} \tag{2.17}$$

$$A(f_i) = \frac{\|(\vec{w}_2 - \vec{w}_1) \times (\vec{w}_3 - \vec{w}_1)\|}{2} \tag{2.18}$$

$$V(f_i) = \frac{A(f_i)\left[\vec{w}_1 \cdot \vec{n}(f_i)\right]}{3} \tag{2.19}$$

The volume, $V(f_i)$, is the volume of the pyramid formed by connecting its three vertices to the origin. Similarly let e_i be an edge delimited by the vertices $\vec{w}_1, \vec{w}_2$ and shared between the two faces g_1, g_2. The midpoint, normal, and length associated with this edge are defined as,

$$\vec{m}(e_i) = \frac{\vec{w}_1 + \vec{w}_2}{2} \tag{2.20}$$

$$\vec{n}(e_i) = \frac{A(g_1)\vec{n}(g_1) + A(g_2)\vec{n}(g_2)}{\|A(g_1)\vec{n}(g_1) + A(g_2)\vec{n}(g_2)\|} \tag{2.21}$$

$$L(e_i) = \|\vec{w}_2 - \vec{w}_1\| \tag{2.22}$$

The normal, $\vec{n}(e_i)$, is the area weighted sum of the normals of the two adjacent faces. We may then define the normal, area, and volume associated with the vertex v_i as weighted sums over the adjacent faces,

$$\vec{n}(v_i) = \frac{\sum_{j=0}^{n_i-1} A(f_{ij})\vec{n}(f_{ij})}{\left\| \sum_{j=0}^{n_i-1} A(f_{ij})\vec{n}(f_{ij}) \right\|} \tag{2.23}$$

$$A(v_i) = \frac{1}{3} \sum_{j=0}^{n_i-1} A(f_{ij}) \tag{2.24}$$

$$V(v_i) = \frac{1}{3} \sum_{j=0}^{n_i-1} V(f_{ij}) \tag{2.25}$$

Lastly, we define the perimeter to be interpolated between two adjacent vertices, v_i and v_{ij},

$$P_{ij} = \left\| \vec{m}(e_{ij}) - \vec{m}(f_{i(j-1)}) \right\| + \left\| \vec{m}(f_{ij}) - \vec{m}(e_{ij}) \right\| \tag{2.26}$$

The fields are defined only on the vertices of the lattice. For brevity we refer to the values of ϕ, H, G on the vertex v_i as ϕ_i, H_i, G_i. To implement the energy functional outlined in the previous section we could compute the values of each term on a discrete lattice. For the simulation it is more useful to compute the local contributions to the energy from each vertex. The global energy can then be computed as a sum over all vertices.

2.4.2 Discrete line tension

To compute the line tension contribution at the vertex v_i we sum up the contributions to the boundary from each vertex and its neighbors,

$$L_i = \sum_{j=0}^{n_i-1} (1 - \delta_{\phi_i \phi_{ij}}) P_{ij} \tag{2.27}$$

where the Kronecker delta ensures that only portions of boundary for which the adjacent vertices have opposite phase are counted. The contour used to interpolate the phase interface

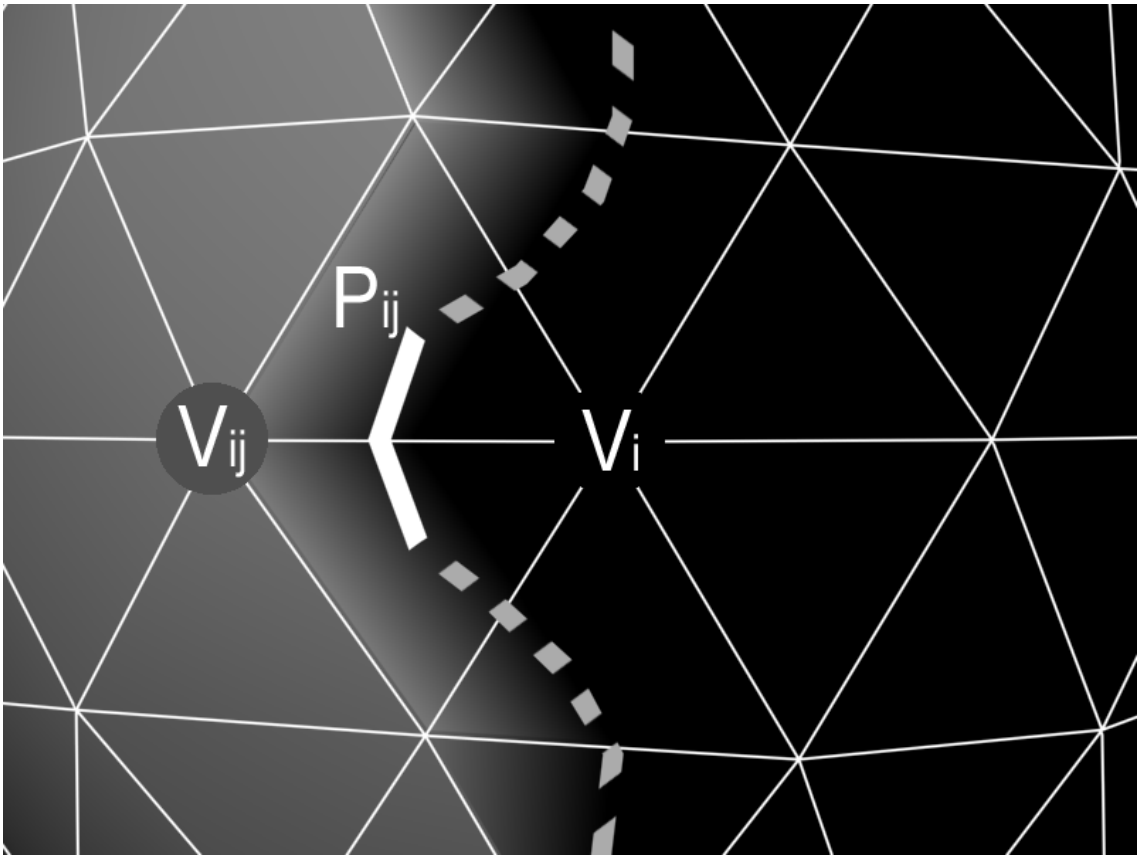


Figure 2.3: Close up of the boundary between the two phases. grey, L_d phase, black, L_o phase. The discrete interpolated boundary is shown dashed. The interpolated boundary between two adjacent vertices (v_i and v_{ij}) is shown in thick white, P_{ij}.

is shown in figure 2.3. The line tension contribution of vertex v_i is then,

$$\mathcal{H}_P(v_i) = \gamma L_i \tag{2.28}$$

2.4.3 Discrete mean curvature

The mean curvature, H_i, is calculated on each vertex using a discrete approximation to the first variation of area formula. The disc-shaped region around the vertex with initial area $A(v_i)$ is projected out a distance ΔR along the normals of the edges, as shown in figure 2.4. Let the two faces $g_{ij}^{(1)}$ and $g_{ij}^{(2)}$ be defined by the three points,

$$g_{ij}^{(1)} \equiv \{\vec{v}_i, \vec{m}(f_{i(j-1)}), \vec{m}(e_{ij})\} \tag{2.29}$$

$$g_{ij}^{(2)} \equiv \{\vec{v}_i, \vec{m}(e_{ij}), \vec{m}(f_{ij})\} \tag{2.30}$$

The projected faces $G_{ij}^{(1)}$ and $G_{ij}^{(2)}$ are given by,

$$G_{ij}^{(1)} \equiv \{\vec{v}_i + \Delta R\vec{n}(v_i), \vec{m}(f_{i(j-1)}) + \Delta R\vec{n}(f_{i(j-1)}), \vec{m}(e_{ij}) + \Delta R\vec{n}(e_{ij})\} \tag{2.31}$$

$$G_{ij}^{(2)} \equiv \{\vec{v}_i + \Delta R\vec{n}(v_i), \vec{m}(e_{ij}) + \Delta R\vec{n}(e_{ij}), \vec{m}(f_{ij}) + \Delta R\vec{n}(f_{ij})\} \tag{2.32}$$

The change in area under projection can then be defined as a sum of the change in area of each face,

$$\Delta A = \sum_{j=0}^{n_i-1} \left[A(g_{ij}^{(1)}) - A(G_{ij}^{(1)}) + A(g_{ij}^{(2)}) - A(G_{ij}^{(2)})\right] \tag{2.33}$$

Now the mean curvature can be defined as,

$$H_i = \frac{\Delta A}{A(v_i)\Delta R} \tag{2.34}$$

The mean curvature term of the energy functional can now be defined on this vertex as,

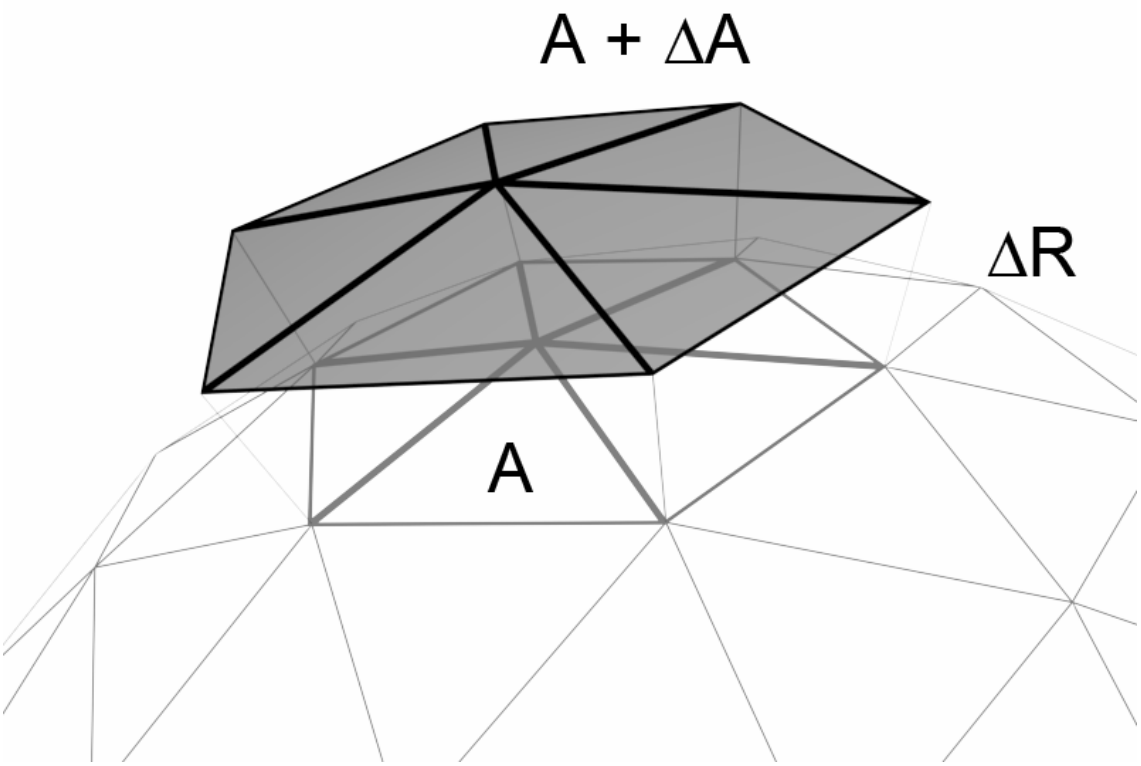


Figure 2.4: The process of mean curvature computation. The initial star-shaped region around a vertex is projected along the surface normals a distance ΔR to produce the shaded grey umbrella region with a difference in area ΔA.

$$\mathcal{H}_H(v_i) = \kappa(\phi_i) H_i^2 A(v_i) \tag{2.35}$$

2.4.4 Discrete Gaussian curvature

The Gaussian curvature field is calculated by taking the angle deficit of the star shaped region around a given vertex [10], v_i, as shown in figure 2.5. With the angle between two edges given by $e_i \angle e_j$, the angle deficit and Gaussian curvature can be written as,

$$\theta_{ij} = \left[e_{ij} \angle e_{i(j+1)} \right] \tag{2.36}$$

$$G_i = \frac{2\pi - \sum_{j=0}^{n_i - 1} \theta_{ij}}{A(v_i)} \tag{2.37}$$

The Gaussian curvature term of the energy functional is now given by,

$$\mathcal{H}_G(v_i) = \bar{\kappa}(\phi_i) G_i A(v_i) \tag{2.38}$$

2.4.5 Additional constraints

In addition to the three terms of our Hamiltonian, we must also include constraint terms that keep the discrete model from taking on impossible configurations. There are five additional constraints we must impose on the lattice:

The areas of individual triangles on the surface are constrained to vary not too far from their initial value. Let $A_0(f_i)$ be the initial area of the face f_i. The local area constraint is,

$$\mathcal{C}_A(f_i) = \begin{cases} 0 & |\Delta A(f_i)| \leq a_l A_0(f_i) \\ \infty & |\Delta A(f_i)| > a_l A_0(f_i) \end{cases}, \tag{2.39}$$

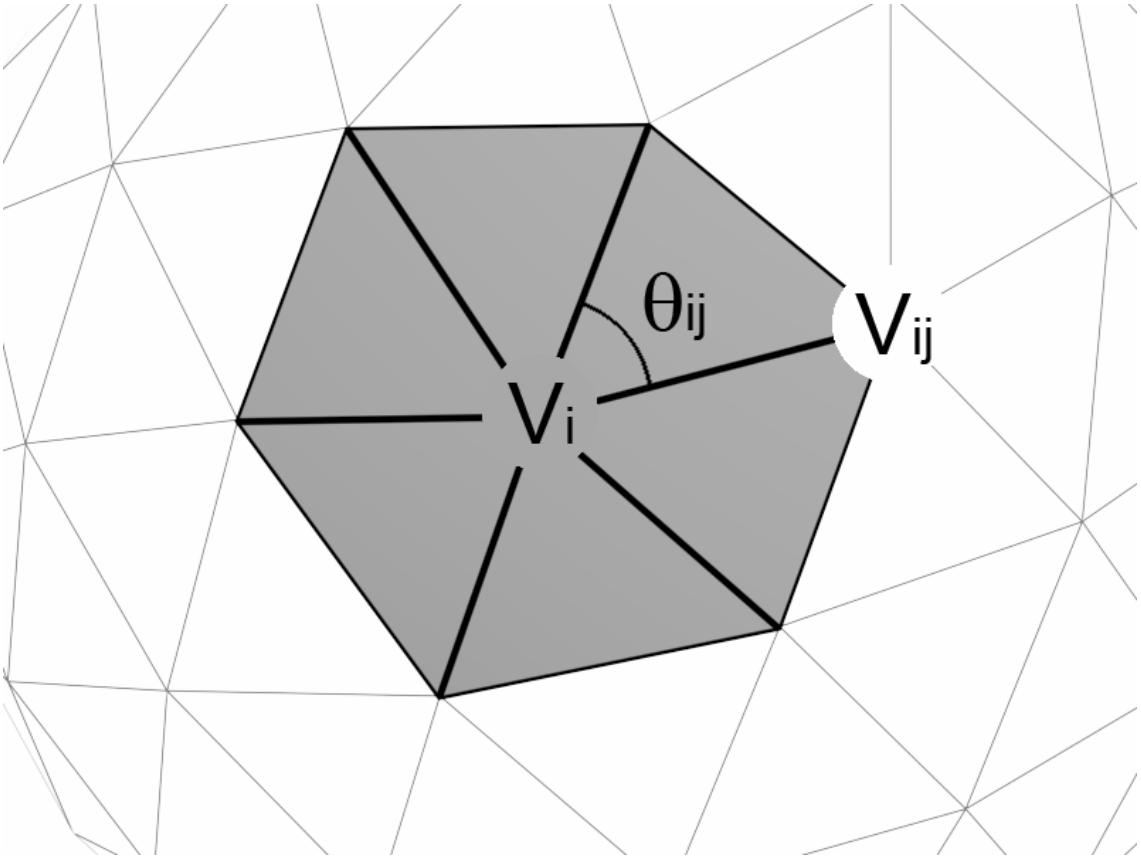


Figure 2.5: Process of Gaussian curvature computation. The angle between adjacent edges is summed around a given vertex to compute the angle deficit.

where $\Delta A(f_i) = A(f_i) - A_0(f_i)$. The parameter a_l gives the maximal wiggle room of the triangles as a fraction of its initial area; $a_l = 0$ implies a rigid lattice.

The length of the edges is also constrained in a similar manner [10]. Let $L_0(e_i)$ be the initial length of a given edge, e_i. The local edge length constraint is,

$$\mathcal{C}_e(e_i) = \begin{cases} 0 & |\Delta L(e_i)| \leq e_l L_0(e_i) \\ \infty & |\Delta L(e_i)| > e_l L_0(e_i) \end{cases} \tag{2.40}$$

where $\Delta L(e_i) = L(e_i) - L_0(e_i)$. Again, the parameter e_l gives a wiggle room as a fraction of the initial length of that edge.

The next constraint is a convexity constraint on the shape of the closed surface. This constrains the normal vector of any face to lie within the 45^o cone about the vector connecting the midpoint to the origin,

$$\mathcal{C}_n(f_i) = \begin{cases} 0 & [\vec{m}(f_i) \angle \vec{n}(f_i)] \leq \pi/4 \\ \infty & [\vec{m}(f_i) \angle \vec{n}(f_i)] > \pi/4 \end{cases} \tag{2.41}$$

This keeps the faces from flipping over and contributing negative volume to the calculations. Notice that each of these constraints are defined on the faces and edges. They can be redefined on the vertices of the lattice as sums over the neighboring faces/edges of that vertex,

$$\mathcal{C}_A(v_i) = \sum_{j=0}^{n_i - 1} \mathcal{C}_A(f_{ij}) \tag{2.42}$$

$$\mathcal{C}_e(v_i) = \sum_{j=0}^{n_i - 1} \mathcal{C}_e(e_{ij}) \tag{2.43}$$

$$\mathcal{C}_n(v_i) = \sum_{j=0}^{n_i-1} \mathcal{C}_n(f_{ij}) \tag{2.44}$$

Next we have the two global constraints,

$$\mathcal{G}_A = A_l \left(A_{tot} - \sum_{i=0}^{N_v-1} A(v_i) \right)^2 \tag{2.45}$$

$$\mathcal{G}_V = V_l \left(V_{tot} - \sum_{i=0}^{N_v-1} V(v_i) \right)^2 \tag{2.46}$$

Both of these constraints serve to keep the total area and volume of the surface close to the fixed values A_{tot} and V_{tot}.

Another hidden constraint is that the total amount of each phase be fixed,

$$\frac{1}{A_{tot}} \sum_{i=0}^{N_v-1} \phi_i = P, \tag{2.47}$$

but this constraint is already imposed in the Monte-Carlo algorithm we use in the simulation. The total energy of the system is summarized as,

$$\mathcal{H} = \mathcal{G}_A + \mathcal{G}_V + \sum_{i=0}^{N_v-1} \left[\frac{1}{2}\mathcal{H}_P(v_i) + \mathcal{H}_H(v_i) + \mathcal{H}_G(v_i) \right] + \sum_{i=0}^{N_v-1} \left[\mathcal{C}_A(v_i) + \mathcal{C}_e(v_i) + \mathcal{C}_n(v_i) \right]. \tag{2.48}$$

The factor of $1/2$ is included in front of $\mathcal{H}_P$ to take into account that each piece of boundary is double counted.

2.4.6 Monte Carlo simulation

The Monte Carlo simulation works in two stages, one corresponding to phase exchanges, and one corresponding to vertex movements [10].

Stage 1: For the phase exchanges we choose two vertices at random, v_i and v_j. If they have

different phases ($\phi_i \neq \phi_j$) then we swap the two phases and calculate the change in energy using the discrete Hamiltonian defined above, ΔE. This procedure is then accepted with the probability $e^{-\Delta E/kT}$. If the move is rejected then the system is returned to the previous configuration. Since phases are only exchanged between vertices, the conservation of total phase is trivially imposed so long as the initial configuration has a specified phase fraction P.

Stage 2: Similarly, for the vertex movements we choose a random vertex v_i and a random unit direction in space $\vec{r}$. We then move the vertex a small distance (about 0.1% of the radius of the GUV) in that direction. All of the curvature and constraint fields are updated and the change in energy, ΔE, due to this perturbation is calculated. It is accepted with probability $e^{-\Delta E/kT}$. If it is rejected then the vertex is moved back to its original position.

This procedure is iterated a large number of times (roughly $10{,}000 \times N_v$ with phase exchanges and vertex movements performed in a 2:1 ratio) and annealed at a linear rate from high temperature until the system has achieved a minimal energy state to within thermal fluctuations. This minimal energy state defines the equilibrium morphology that we would expect to observe in experiments where GUVs have had adequate time to anneal.

2.5 Results and Discussion

2.5.1 Curvature and line tension can produce modulated phases.

Implementing the discrete model outlined above yielded the appearance of modulated phases similar to those observed experimentally on GUVs in this four-component system. The simulation shows a strong periodic pattern arising from the competition between the line tension and curvature terms. Shown in Table 2.1 is the canonical parameter set that gives

parameter	value	unit
γ	0.01	pN
κ_d	10×10^{-19}	J
κ_o	100×10^{-19}	J
$\bar{\kappa}_d$	-10×10^{-19}	J
$\bar{\kappa}_o$	-100×10^{-19}	J
R	25	μm
P	0.5	N/A

Table 2.1: Canonical parameter set that is found to produce modulated phases.

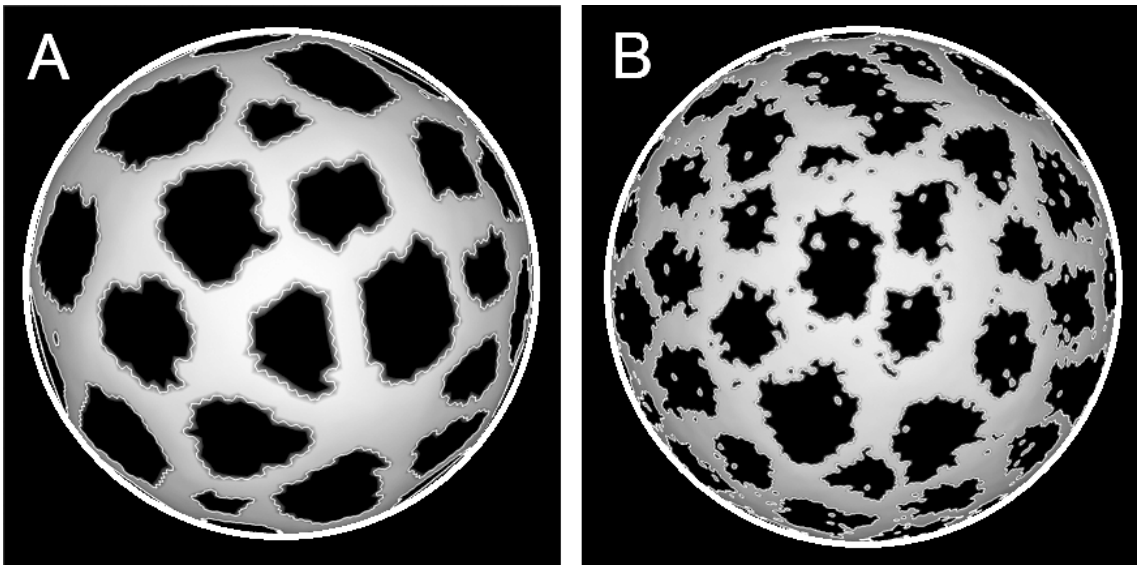


Figure 2.6: Modulated phases can be simulated. Comparison of the pattern on a low resolution (10,000 vertex) lattice (A) to the pattern on a high resolution (40,000 vertex) lattice (B) shows that the patterning is not an artifact of the triangulation.

rise to a strong modulated phase pattern.

The parameters shown here were found by trial and error search of the parameter space. The bending moduli here are about one order of magnitude higher than the experimental values reported for some (different) lipid mixtures. The reported values put the typical bending moduli at values of around $\kappa_d = 2 \times 10^{-19}$ J and $\kappa_o = 8 \times 10^{-19}$ J [21, 18, 4]. Notice that we have set $\kappa \approx -\bar{\kappa}$ for both the L_d and L_o phases in accordance with the observation that $\bar{\kappa}/\kappa \approx -0.9$ [11].

The line tension reported here is two orders of magnitude below typical experimentally reported values, $\gamma \approx 1.2$ pN [21, 4]. Line tension values in our simulation of modulated phases should be smaller than those measured experimentally because the values of line tension reported in the literature correspond to large macroscopic domains (the shape and stability of macroscopic domains being the principle method of extracting this parameter).

We identify two reasons for this seeming discrepancy: First, line tension cannot be measured by any method yet reported in the regime of modulated phases. With near-certainty we can say that it must be much lower for such mixtures; Second, given that we used only the bending energies as the interactions that oppose line tension, then if additional interactions are in play besides these bending energies, the line tension to produce modulated phase morphology would be larger. We are considering other potential long range interactions (such as dipolar repulsion arising from the molecular structure of the lipids) to include in future simulations to test this hypothesis.

In order to ensure that the modulated phases are not an artifact in the simulation we ran the simulation on much higher resolution lattices to see if the patterns persist. Figure 2.6 shows a side-by-side view of the patterns on a lattice containing $10,000$ vertices and one containing $40,000$ with the same parameter set. The poor scaling of the simulation with vertex number makes examination of much higher resolution problematic.

Figure 2.7 shows that with the same parameters as above, but with $P = 0.75$, simulated patterns very closely match experimental patterns observed on GUVs.

2.5.2 Modulated phases are thermodynamically stable

One important issue about modulated phases is whether or not the patterns are thermodynamically stable or instead are a kinetically trapped state. The minimal energy morphology of a phase-separated GUV in the absence of curvature is known to be a single round macro-

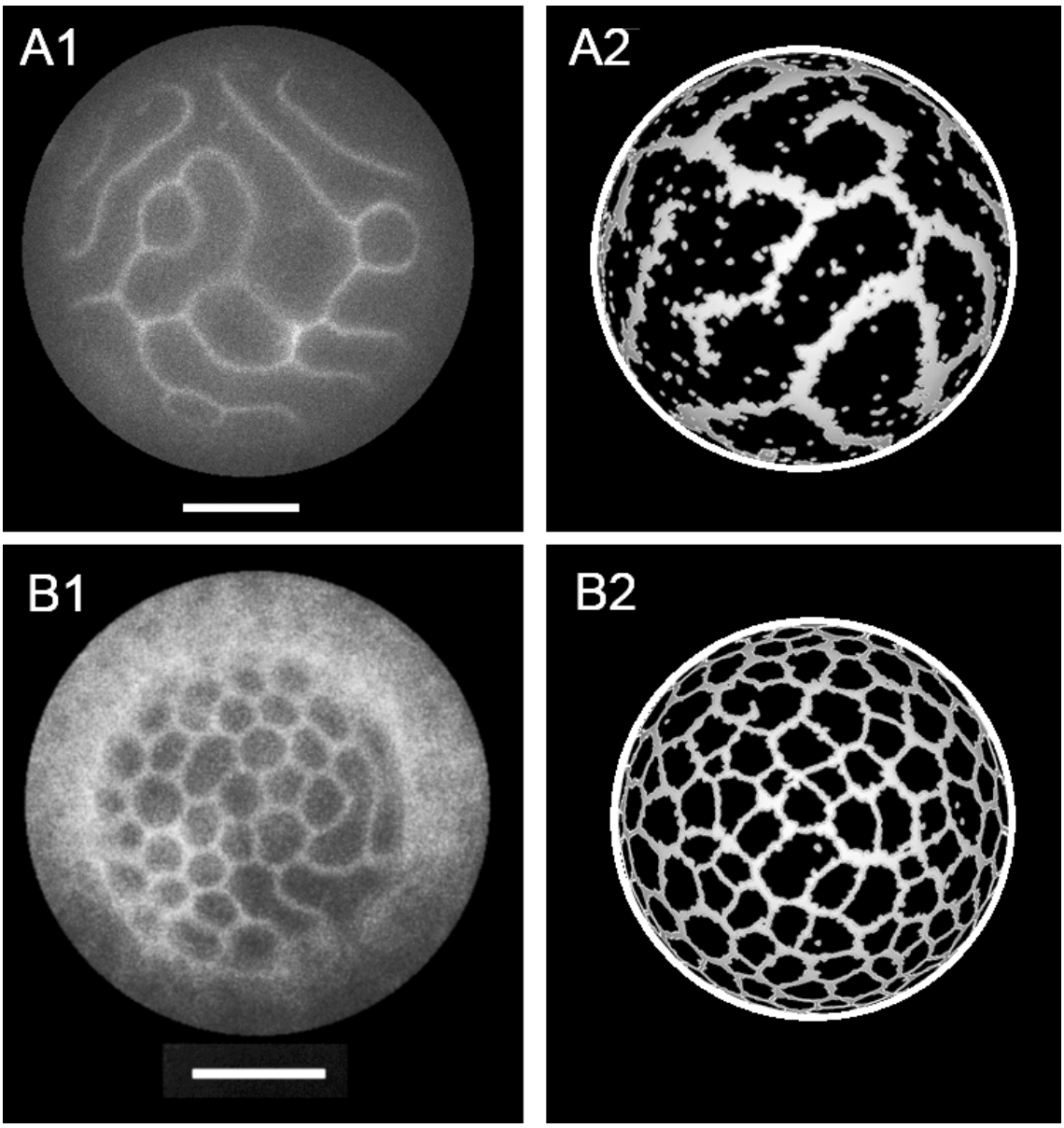


Figure 2.7: Direct comparison of modulated phases observed experimentally on the surface of GUVs to patterns produced in simulation. (A) Simulation with $P = 0.75$ (A2) showing stripe-like patterns similar to those on an actual GUV (A1). (B) Simulation with $P = 0.75$ and $\kappa_o = -\bar{\kappa}_o = 300 \times 10^{-19}$J showing honeycomb patterns (B2). GUV in (A1) was taken from figure 1(C) for comparison with simulated GUV (A2). GUV composition (DSPC/DOPC/POPC/CHOL) in (B1) is 0.45/0.09/0.21/0.25. C12:0 DiI (0.02 mol%) was used in both (A1) and (B1). Scale bars 10 μm, temperature 23C

scopic domain [2], minimizing the perimeter-to-area ratio of the domain.

To determine the thermodynamic stability of the modulated phases we compared the energy of a single round domain morphology to that of a modulated phase morphology. We first artificially generated a GUV with a single round domain by turning off the shape exchange portion of our Monte Carlo simulation, then allowing the GUV to equilibrate. This gave a single round domain, but did not contain the energetic contributions of the thermal fluctuations. To capture the thermal fluctuations we performed a second stage equilibration where phase exchanges were turned off and the shape exchanges were allowed to equilibrate. As figure 2.8 shows, the energy of the single round domain morphology exceeds that of the modulated phase morphology indicating that the modulated phase morphology is more thermodynamically stable.

All of the simulations shown here were started from a melted state (vertex phases randomly dispersed). A second test of the stability of the modulated phases was to start the system off in a single large domain state and then allow the system to relax to see if it still transitions to a modulated phase morphology. This process is shown in figure 2.9.

Starting from a macroscopically phase-separated GUV, a transition to modulated phases appears, again indicating that the modulated phases are thermodynamically more stable than the single round domain. Notice that the energy increases at the beginning of the simulation, seemingly in contradiction of the second law of thermodynamics. This is because we have plotted only the enthalpic part of the free energy, neglecting the increase in entropy as the domain begins to break apart.

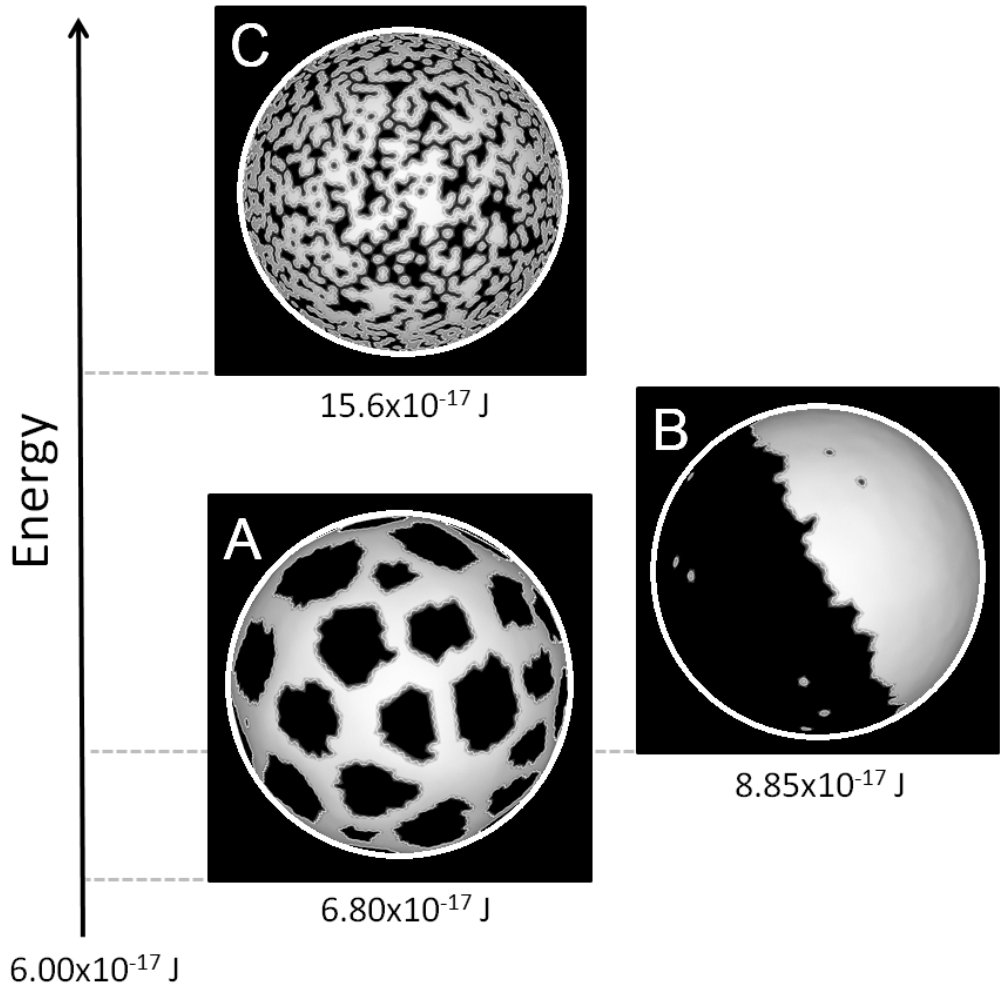


Figure 2.8: With the parameters given in Table 2.1, the modulated phase morphology (A) is the lowest energy state. The single round domain mophology (B) has higher energy, and the melted lattice (C) has the highest energy and most phase boundary.

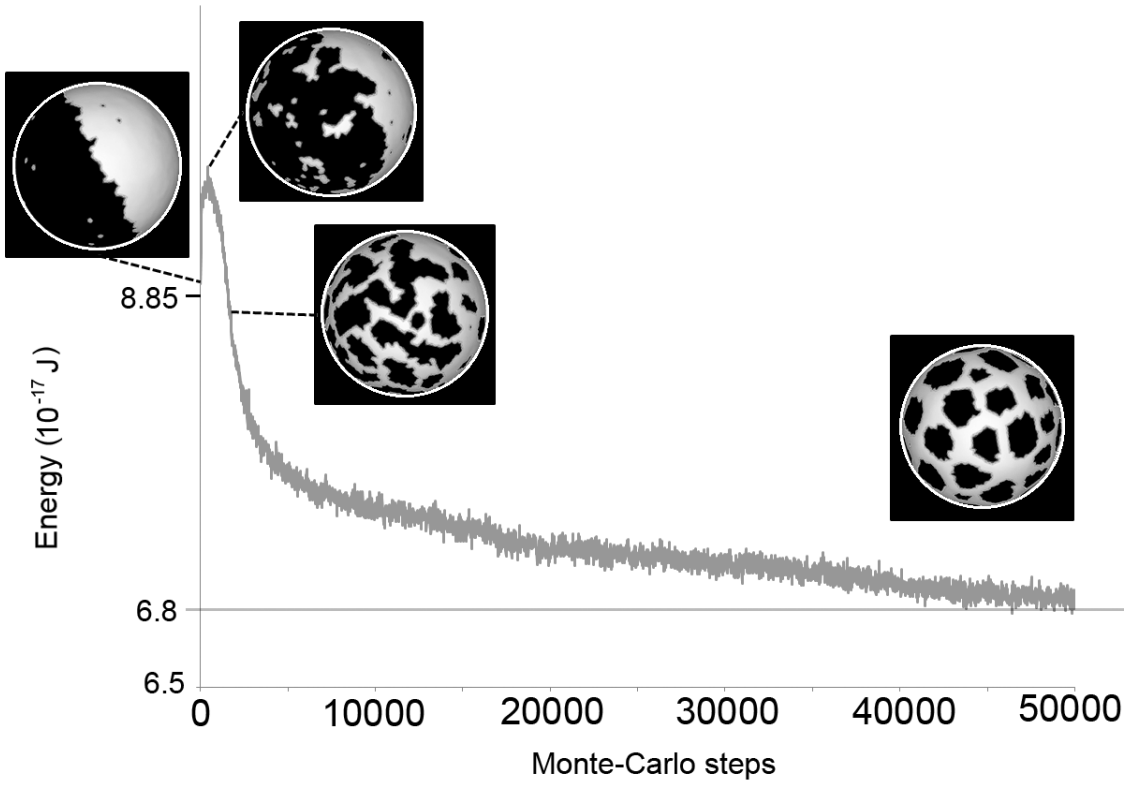


Figure 2.9: Macroscopic domains transition to modulated phases. Energy as function of Monte-Carlo steps starting from a single macroscopic domain, which breaks up and eventually stabilizes to the modulated phase pattern. The simulation can take a long time to equilibrate as shown by the continually decreasing energy, but the pattern does not change appreciably in the tail end of the curve. The slight increase in energy at the beginning is probably compensated by the increase in entropy as the domain boundary begins to disintegrate.

2.5.3 The physical parameters of the phases and the GUV dictate the phase morphology.

The simulation model shown here has seven tunable parameters, each affecting the energy landscape in some complex way. It is useful to do some exploration of how these parameters change the nature of the morphology. Moving along a tie-line at a fixed value of ρ will only change the amount of each phase (P) on the surface without affecting any of the other parameters. This process is shown in figure 2.10 for three different values of P.

The line tension, γ, has a large effect on the phase morphology. The curvature and the line tension terms work in tandem to produce the modulated phases. If this balance is perturbed the system abruptly transitions to a melted or macroscopically phase-separated morphology, figure 2.11.

At high line tension we noticed a large degree of kinetic trapping caused by the very large energy barrier to a single phase exchange. This caused non-equilibrium stabilization of multiple small domains. Kinetic trapping of domains by curvature is an experimentally observed phenomenon [26], but we are interested in studying equilibrium configurations.

The difference in bending rigidity between the L_o and L_d controls the width of the L_d lines and the size of the L_o domains. Figure 2.12 shows a series of increasing bending rigidity of the L_o phase while keeping moduli of the L_d phase fixed.

All of the simulations shown here are performed on $R = 25$ μm GUVs. What affect, if any, does the size of the GUV have on the nature of the patterns? We find that scaling the radius of the GUV is energetically equivalent to scaling the line tension,

$$\mathcal{H}[R, \gamma] = \gamma L + \int_S \kappa(\phi) H^2 + \bar{\kappa}(\phi) G dA. \tag{2.49}$$

Scaling R by a factor a will scale all linear quantities by the same factor and all area quantities by a^2. Note that under this scaling the area fraction of each phase, P, is held fixed, so the

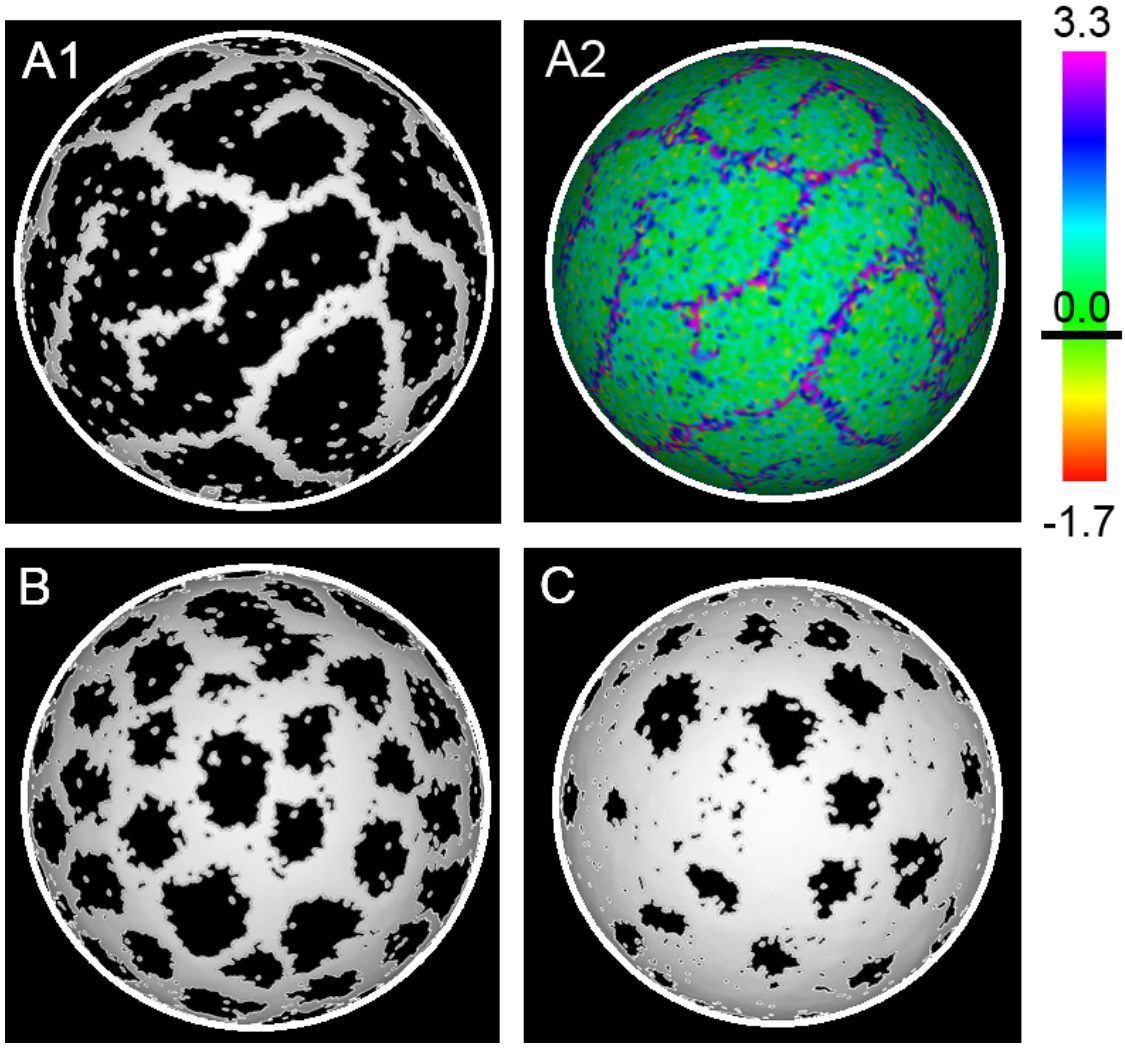


Figure 2.10: (Color) Phase morphologies change as the phase fraction (P) is varied. (A1) $P = 0.75$; (A2) Mean curvature map showing how the modulated phases and the geometry of the membrane couple. Curvature units in the color scale are μm^{-1}; (B) $P = 0.5$; (C) $P = 0.25$. The single pixel domains are not static structures, but transiently flicker in and out of existence due to thermal fluctuations of the membrane shape in the L_o regions.

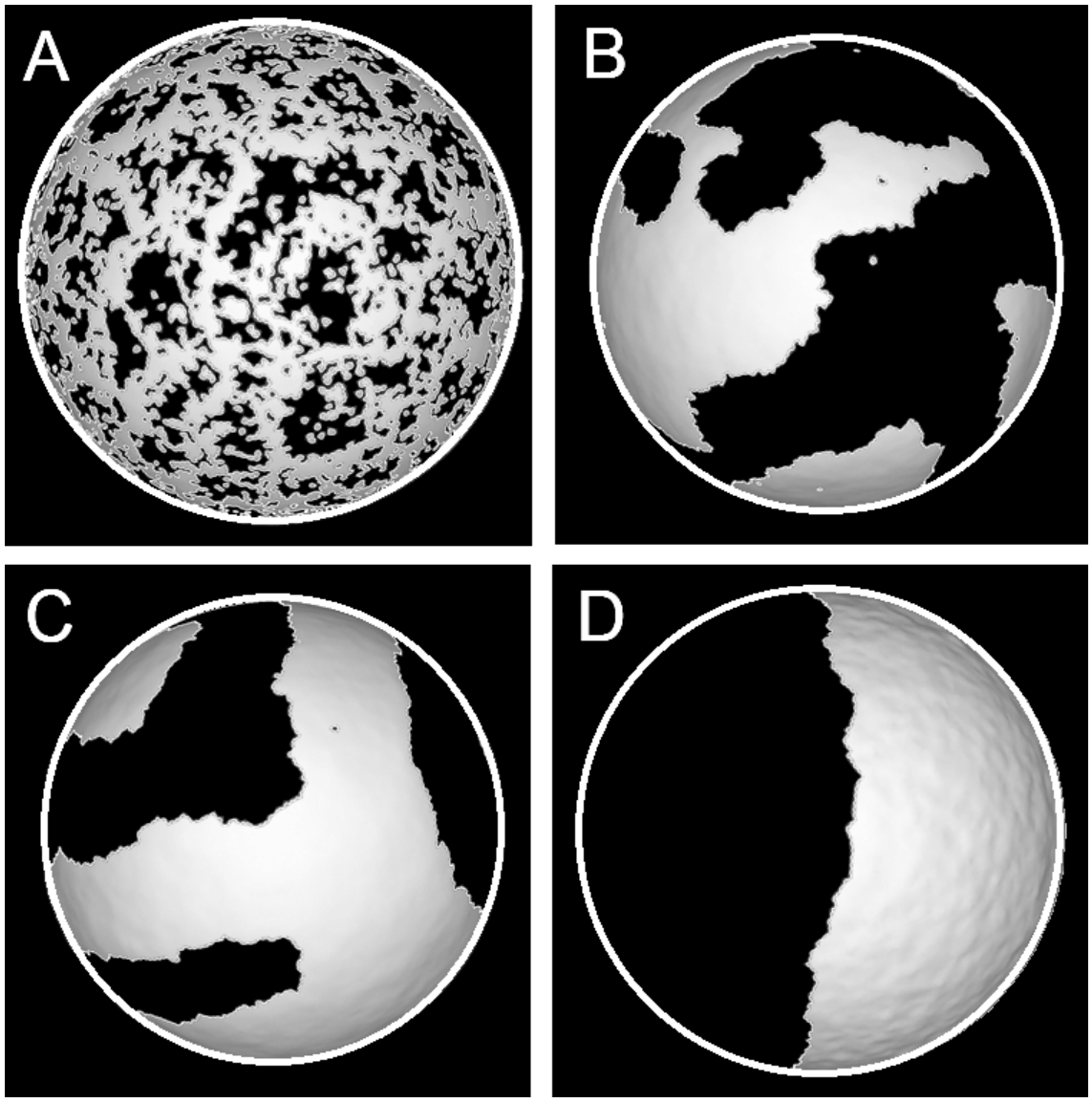


Figure 2.11: Phase morphologies change as line tension is varied. (A) $\gamma = 0.005$ pN. (B) $\gamma = 0.02$ pN. (C) $\gamma = 0.03$ pN. (D) $\gamma = 1.5$ pN.

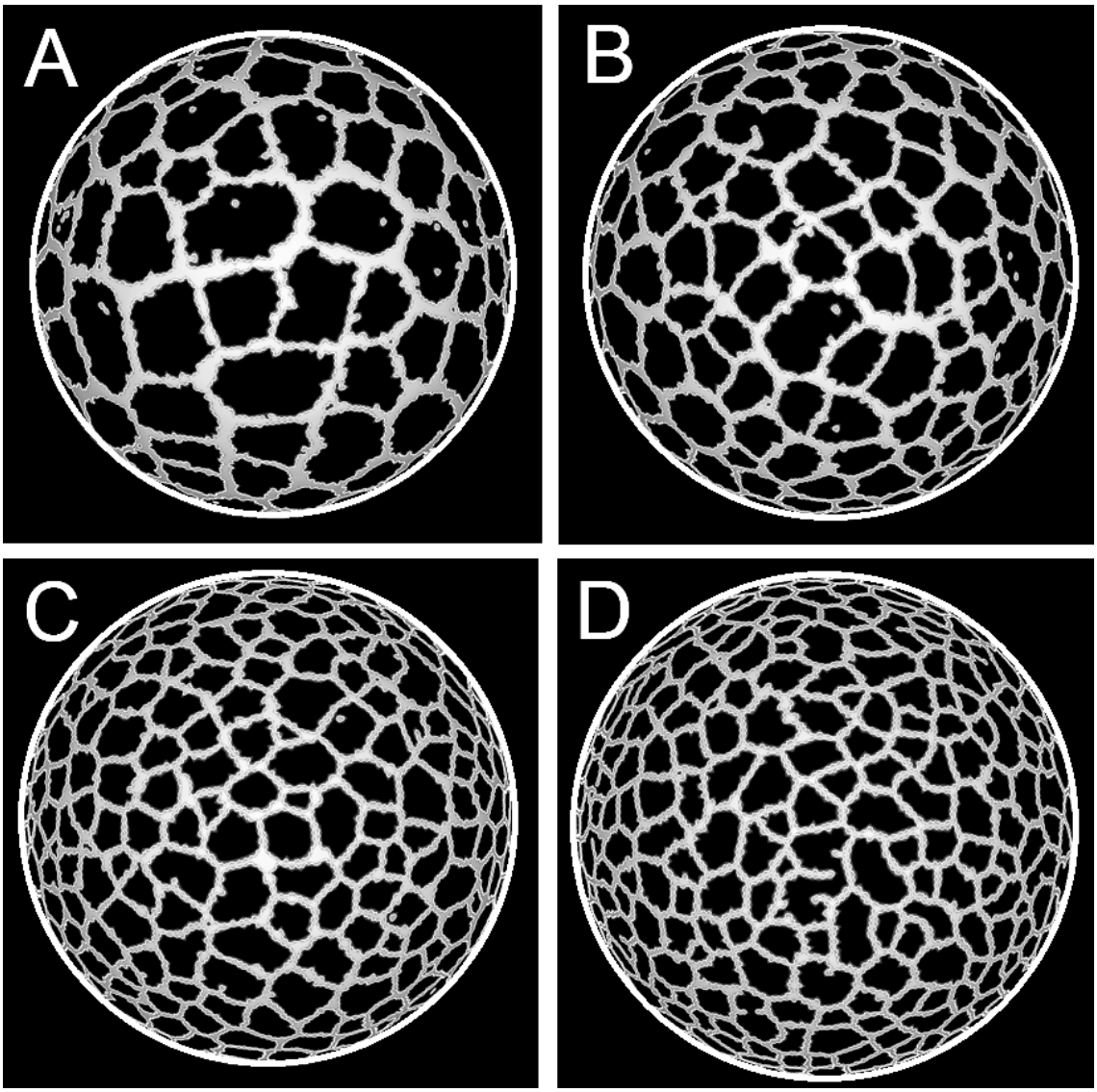


Figure 2.12: Phase morphologies change as the bending moduli (κ_o and $\bar{\kappa}_o$) of the L_o phase are varied. (A) $\kappa_o = -\bar{\kappa}_o = 200 \times 10^{-19}$J. (B) $\kappa_o = -\bar{\kappa}_o = 300 \times 10^{-19}$J. (C) $\kappa_o = -\bar{\kappa}_o = 500 \times 10^{-19}$J. (D) $\kappa_o = -\bar{\kappa}_o = 1000 \times 10^{-19}$J.

actual area of each phase must change accordingly (by a factor of a^2).

$$\mathcal{H}[aR, \gamma] = \gamma(aL) + \int_S \kappa(\phi)(H/a)^2 + \bar{\kappa}(\phi)(G/a^2)(a^2 dA) \tag{2.50}$$

$$= \gamma(aL) + \int_S \kappa(\phi)H^2 + \bar{\kappa}(\phi)GdA \tag{2.51}$$

$$= \mathcal{H}[R, a\gamma] \tag{2.52}$$

Thus scaling the GUV size is equivalent to scaling the line tension, as in figure 2.11. See Supplemental Material for a detailed proof of these scaling relations [1].

2.6 Conclusion and Future Directions

We have shown that the Helfrich model can explain the existence of modulated phase patterning: Patterns obtained through simulation closely match those experimentally observed on the surfaces of GUVs in the four-component system DSPC/DOPC/POPC/CHOL and represent thermodynamically stable configurations. The parameters of line tension and bending energies yield a range of different patterns (including striped, honeycomb, and macroscopic) depending on their relative and absolute values. This successful modeling provides a foundation for systematic study of the control of domain size in phase-separated biological membranes.

2.6.1 Future Studies

1. Spherical topology, can break the symmetry of the curvature fields. By performing simulations on flat planar lattices, we will examine if patterning persists.

2. Accurate measurements of the various model parameters are central to relating our

simulations to the experimental system. We plan to use the microscopy-based method used by Semrau et al. [21] to measure elastic parameters as a function of ρ in our four-component system. These experimentally-determined parameters will enable us to test the validity and accuracy of the simulation in faithfully representing the energetics of phase separation in our four-component bilayer mixture.

3. Membrane tension might have a strong influence on modulated phases. In all of the simulations shown here the global volume constraint, $\mathcal{G}_V$, is omitted to allow water to equilibrate across the membrane. Including a global volume constraint and setting the equilibrium volume, V_{tot}, lower or higher than that of a uniform sphere, enables simulation of the changes in membrane tension caused by osmotic stress. We plan to compare such simulations with an experimental system where GUVs are subjected to a change in osmotic pressure across the membrane.

4. Our model makes the strong prediction that the ratio of vesicle diameter to line tension is the key parameter, along with bending energies, that control domain morphology. Studies are underway to verify the size and line tension scaling relation by correlating phase morphology with vesicle size, both experimentally and in simulations.

5. The phase coexistence itself, as well as line tension and bending moduli, are sensitive to temperature. As a further study of modulated phase equilibrium, we will compare the temperature dependence found in the simulations with experimental observations of the temperature-controlled GUVs on the microscope stage.

2.7 Acknowledgements

Support was from a research award from NSF MCB 0842839 (to G.W.F.). J.J.A. received support from NIH Training Grant award 1-T32 GM 08267.

Bibliography

[1] See Supplemental Material at [URL will be inserted by publisher] for a detailed proof of these scaling relations.

[2] S.A. Akimov, P.I. Kuzmin, J. Zimmerberg, and F.S. Cohen. Lateral tension increases the line tension between two domains in a lipid bilayer membrane. *Phys. Rev. E.*, 75: 011919, 2007.

[3] G Arreaga, R. Capovilla, C. Chryssomalakos, and J. Guven. Area-constrained planar elastica. *Phys. Rev. E.*, 65:031801, 2002.

[4] T. Baumgart, S. Hess, and W. Webb. Imaging coexisting fluid domains in biomembrane models coupling curvature and line tension. *Nature*, 425:821–824, 2003.

[5] F. Brown. Elastic modeling of biomembranes and lipid bilayers. *Annu. Rev. Phys. Chem.*, 59:685–712, 2008.

[6] J. Fan, M. Sammalkorpi, and M. Haataja. Influence of nonequilibirum lipid transport, membrane compartmentalization, and membrane proteins on the lateral organization of the plasma membrane. *Phys. Rev. E.*, 81:011908, 2010.

[7] A. Hammond, F. Heberle, T. Baumgart, D. Holowka, B. Baird, and G. Feigenson. Crosslinking a lipid raft component triggers liquid ordered-liquid disordered phase separation in model plasma membranes. *PNAS*, 102:6320–6325, 2005.

[8] F. Heberle, J. Wu, S. Goh, R. Petruzielo, and G. Feigenson. Comparison of three ternary lipid bilayer mixtures: Fret and esr reveal nanodomains. *Biophysical Journal*, 99:3309–3318, 2010.

[9] F. Heberle, R. Petruzielo, J. Pan, P. Drazba, N. Kucerka, R. Standaert, G. Feigenson, and J. Katsaras. Lipid acyl chain unsaturation controls raft size in model membranes. *Submitted*, 2012.

[10] J. Hu, T. Weikl, and R. Lipowsky. Vesicles with multiple membrane domains. *Soft Matter*, 7:6092, 2011.

[11] M. Hu, J. Briguglio, and M. Deserno. Determining the gaussian curvature modulus of lipid membranes in simulations. *Biophysical Journal*, 102:1403–1410, 2012.

[12] A. Imperio and L. Reatto. Microphase separation in two-dimensional systems with competing interactions. *J. Chem. Phys.*, 124:164712, 2006.

[13] Y. Kaizuka and J. Groves. Bending-mediated superstructural organizations in phase-separated lipid membranes. *New J. of Phys.*, 12:095001, 2010.

[14] P. Kingsley and G. Feigenson. The synthesis of a predeuterated phospholipid: 1,2-dimyristoyl-sn-glycero-3-phosphocholine-d72. *Chem. and Phys. of Lipids*, 24:135–147, 1979.

[15] T. Konyakhina, S. Goh, J. Amazon, F. Heberle, J. Wu, and G. Feigenson. Control of nanoscopic-to-macroscopic transition: modulated phases in four-component dspc/dopc/popc/chol giant unilamellar vesicles. *Biophysical Journal*, 101:L08–L10, 2011.

[16] D. Lingwood and K. Simons. Lipid rafts as a membrane-organizing principle. *Science*, 327:46–50, 2010.

[17] J. Liu, S. Qi, J. Groves, and A. Chakraborty. Phase segregation on different length scales in a model membrane system. *J. Phys. Chem. B*, 109:19960–19969, 2005.

[18] D. Marsh. Elastic curvature constants of lipid monolayers and bilayers. *Chem. and Phys. of Lipids*, 144:146–159, 2006.

[19] J. Pecreaux, H. Dobereiner, J. Prost, J. Joanny, and P. Bassereau. Refined contour analysis of giant unilamellar vesicles. *Eur. Phys. J. E*, 13:277–290, 2004.

[20] S. Rozovsky, Y. Kaizuka, and J. Groves. Formation and spatio-temporal evolution of periodic structures in lipid bilayers. *J. Am. Chem. Soc.*, 127:36–37, 2005.

[21] S. Semrau, T. Idema, L. Holtzer, T. Schmidt, and C. Storm. Accurate determination of elastic parameters for multicomponent membranes. *Phys. Rev. Lett.*, 100:088101, 2008.

[22] P. Sengupta, D. Holowka, and B. Baird. Flourescence resonance energy transfer between lipid probes detects nanoscopic heterogeneity in the plasma membrane of live cells. *Biophysical Journal*, 92:3564–3574, 2007.

[23] M. Seul and D. Andelman. Domain shape and patterns: the phenomenology of modulated phases. *Science*, 267:476–483, 1995.

[24] M. Swamy, L. Ciani, M. Ge, A. Smith, D. Holowka, B. Baird, and J. Freed. Coexisting domains in the plasma membranes of live cells characterized by spin-label esr spectroscopy. *Biophysical Journal*, 90:4452–4465, 2006.

[25] A. Travesset. Effect of dipolar moments in domain size of lipid bilayers and monolayers. *J. Chem. Phys.*, 125:084905, 2006.

[26] T. Ursell, W. Klug, and R. Phillips. Morphology and interaction between lipid domains. *PNAS*, 106:13301–13306, 2009.

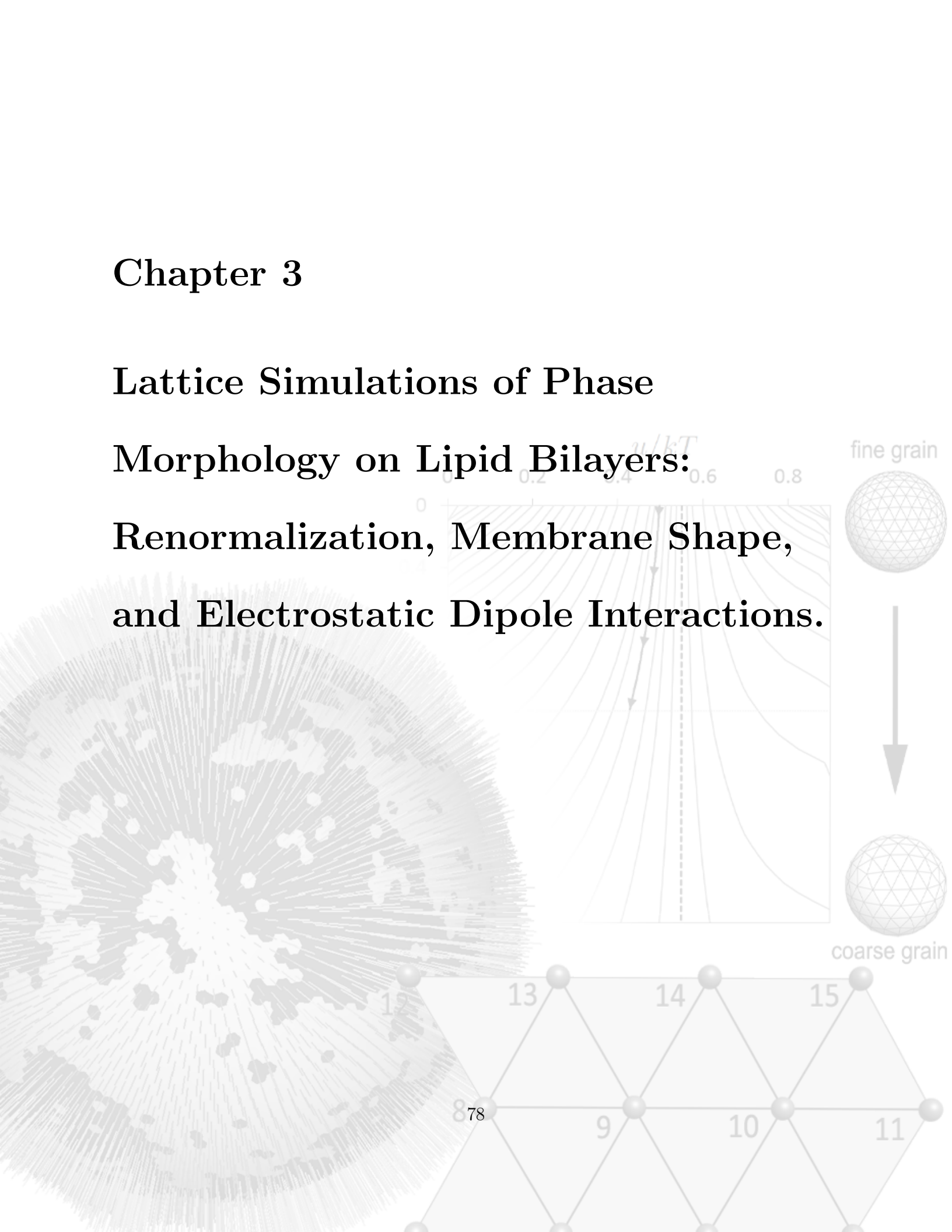
[27] S. Veatch and S. Keller. Seeing spots: complex behavior in simple membranes. *Biochim. Biophys. Acta.*, 1746:172–185, 2005.

[28] Y. Yoon, J. Hale, P. Petrov, and P. Cicuta. Mechanical properties of ternary lipid membranes near a liquid-liquid separation boundary. *J. Phys. Condens. Matter*, 22: 062101, 2010.

[29] J. Zhao, J. Wu, F. Heberle, T. Mills, P. Klawitter, G. Huang, G. Costanza, and G. Feigenson. Phase studies of model biomembranes: complex behavior of dspc/dopc/cholesterol. *Biochim. Biophys. Acta.*, 1768:2764–2776, 2007.

Chapter 3

Lattice Simulations of Phase Morphology on Lipid Bilayers: Renormalization, Membrane Shape, and Electrostatic Dipole Interactions.



3.1 Abstract

When liquid phases coexist at equilibrium but are not driven to minimize domain interfacial contact energy, the resulting patterns of phase domains can have important implications for living cells. In this study we explore some of the interactions and conditions that produce the stable patterned phases that are observed in model lipid mixtures. By use of Monte Carlo simulations we find that background curvature is important for the formation of patterned ("modulated") phases. The interactions that stabilize nanoscopic phase separation are still not well understood. We show that inclusion of an electrostatic dipole repulsion with decay lengths as short as 2-4 lipid diameters can break up domains at the nanometer scale and that the location of the miscibility critical point is sensitive to this interaction. The use of a coarse-grained simulation raises questions about comparing parameters in simulations performed at different length scales. Using renormalization group techniques we show how to reconcile this problem, treating line tension as a running coupling constant.

3.2 Introduction

Lipid rafts are nanoscopic phase domains thought to exist in plasma membranes [40, 35, 28, 23]. Rafts might be important for the cell to control protein localization, signaling, exocytosis, endocytosis, and have been implicated in virus assembly on the inner leaflet [7]. While the importance of lipid rafts is clear, the microscopic details of their formation and stability is still a question of active research.

The plasma membrane is composed of hundreds of individual lipid and protein species. To study the nature of lipid rafts, truncated model systems are employed that limit the complexity to just a handful of lipids. As few as three lipid species are sufficient to produce liquid-liquid ($L_d + L_o$) coexisting phases that are a visible manifestation of the domains pos-

tulated by the raft model of the plasma membrane [22, 37, 32]. Depending on the lipids used in model system studies, a wide range of phase morphologies can be realized. The system distearoylphosphatidylcholine (DSPC)/dioleoylphosphatidylcholine (DOPC)/cholesterol (chol) can exhibit macroscopic domains tens of microns in diameter, readily observable by optical fluorescence microscopy of giant unilamellar vesicles (GUV) [46]. By contrast, the lipid system DSPC/palmitoyl,oleoyl-phosphatidylcholine (POPC)/chol appears uniform in GUV studies, but a large body of evidence, employing sub-optical techniques such as Forster resonance energy transfer (FRET), electron spin resonance (ESR), and small angle neutron scattering (SANS), supports the presence of coexisting phase domains at nanometer scales [3, 11, 12, 8].

The nanodomains present in model systems are thought to be similar in nature to the lipid rafts of living cells [4, 38]. Thus a thorough theoretical and empirical exploration of these model systems would clarify a fundamental biological phenomenon. To study the nature of this nanoscopic phase separation, we previously used a 4-component lipid mixture to enable gradual, composition-induced transition from GUVs exhibiting macroscopic phase domains (DSPC/DOPC/chol) to GUVs with nanodomains (DSPC/POPC/chol) [9, 21]. This experiment revealed a narrow range of compositions that exhibits modulated phase behavior (patterned phases that are periodic and thermodynamically stable). Our previous study [2] found that modulated phases could be explained by a competing interactions model [36, 19, 18]: line tension, which drives coexisting phases towards large round domains, competes against curvature energies, which penalize bending of the membrane.

This work expands the explorations of this model. We find that the line tension used to describe energy contributions from the phase boundary is highly sensitive to coarse-graining. A renormalization group approach is used to formalize this problem and provide a workable solution. Renormalization was crucial for comparing simulations performed at different grain levels and for connecting our simulation results directly to experimental observations.

We also explore the role of background curvature in the stabilization of modulated phases on the surface of GUVs. Whereas all simulations in the previous study were performed on spherical lattices, which break the symmetry of the Hamiltonian and may facilitate patterning of the phases, here we construct lattices that lack this background curvature and compare the resultant morphologies for similar parameter sets.

With the success of a competing interactions model to explain modulated phases, we explore the use of electrostatic dipole repulsion between lipids as another potential interaction to compete with line tension to stabilize nanodomains. We study the effects of the dipole density in each phase and the decay length of the electric fields within the bilayer on phase morphology. Simulations are performed to study limited inclusion of curvature as an added degree of complexity. These results are compared directly to the domain size measurements of Heberle et al. [12] using SANS on 60nm large unilamellar vesicles(LUV).

3.3 Materials and Methods

3.3.1 Monte Carlo Simulation

We use a triangulated lattice to model the membrane and perform a Monte Carlo simulation to minimize a Hamiltonian defined on the vertices [2, 16]. The perturbations explored at each step depend on the fields being studied. The Hamiltonian we had previously studied, the Helfrich Energy functional [13], models the energetics of an elastic membrane, S,

$$\mathcal{H}[S,\phi] = \gamma L + \iint_S \kappa(\phi)[H]^2 dA + \iint_S \bar{\kappa}(\phi) G dA \tag{3.1}$$

where γ is the line tension, L is the total length of phase boundary, κ is the bending modulus, and $\bar{\kappa}$ is the saddle-splay modulus. The parameters used in each section are defined in Tables 3.1 and 3.2. The fields here are the phase field, $\phi(\vec{r})$ (a binary field taking on the

value 0 in L_d phase and 1 in L_o phase), the mean curvature field $H(\vec{r})$, and the Gaussian curvature field $G(\vec{r})$. To perturb the phase field ϕ, long range exchanges are performed in which the phases of two randomly chosen vertices are exchanged. To perturb the curvature fields, a random vertex is moved a small distance (0.1% of the vesicle radius) in a random direction [16]. These two procedures enable the simulation to explore a large range of phase morphologies and vesicle shape deformations. Accepting or rejecting these perturbations using the Metropolis-Hastings criterion, with probability $P = e^{-\Delta E/kT}$, biases the simulation towards a minimal energy configuration, but still allows for entropic fluctuations.

Some key shape constraints prevent the surface from becoming non-physical. Edge lengths are constrained so that each edge is free to fluctuate within $\pm 30\%$ of a specified value (here we use the average of the initial edge lengths). This keeps the lattice from self-intersecting while still allowing a reasonable degree of flexibility. The global area of the lattice is also harmonically constrained so that the total area of the membrane fluctuates by $< 1\%$. This constraint reflects the relatively high compression modulus of the membrane and keeps area fluctuations small. The simulation framework outlined above is exactly the same as in our previous work [2]. The changes introduced below are new to this study and extend the applicability of the simulation to more general cases.

3.3.2 Open Surfaces and Flat Topology

The surface representation was changed to facilitate open surfaces and edges. This enables study of the behavior of flat open sheets of membrane and comparison of their behavior to that of curved membranes. An open surface can be modeled as a closed surface in which a subset of the vertices have missing neighbors. In the previous formalism we imposed the restriction that the array of neighbors for every vertex must form a closed counter-clockwise loop. Here we lift this constraint and represent the missing points as a special

value, internally referred to as 'BREAK', in the neighbor array. An example of the neighbor array for a vertex on the edge of a surface is shown in figure 3.1. The occurrence of a break signals the simulation to use edge case versions of the geometric update routines.

Constructing the flat lattice requires defining the positions and neighbor arrays of each vertex. For an $N \times N$ flat array the position of the i^{th} vertex is,

$$\vec{v}_i = \left\langle \frac{(i\%N)}{N} + \frac{\lfloor i/N \rfloor \%2}{2N} - \frac{1}{2}, \frac{\sqrt{3}}{2}\left(\frac{\lfloor i/N \rfloor}{N} - \frac{1}{2}\right), 0 \right\rangle, \tag{3.2}$$

where "$\cdot\%\cdot$" is the modulo function and $\lfloor \cdot \rfloor$ is the floor function. This produces the triangular lattice grid shown in figure 3.2. The connectivity of this lattice is defined by the edges shown, with 'BREAK' flags inserted where appropriate. The boundary conditions for the open surface hold the vertices at the edges fixed in space. This constraint keeps the surface from folding into itself, which was observed when the boundary points were allowed to move.

3.3.3 Gaussian Curvature on Open Surfaces

The inclusion of open surfaces requires a re-definition of the way Gaussian curvature is calculated on a discrete lattice. Gaussian curvature can be calculated by summing the angles between neighboring edges around a given vertex and subtracting this from 2π to determine the angle deficit [16, 2]. This approach cannot be applied directly when the neighbors do not form a complete loop.

To generalize the computation of Gaussian curvature to vertices that could be adjacent to an edge, the fan formed by the faces around that vertex is projected onto the tangent plane at that vertex. The angle represented by the two edges adjacent to the break, θ_P, is then measured. This geometric construction is shown in figure 3.3. The equation for Gaussian curvature then becomes,

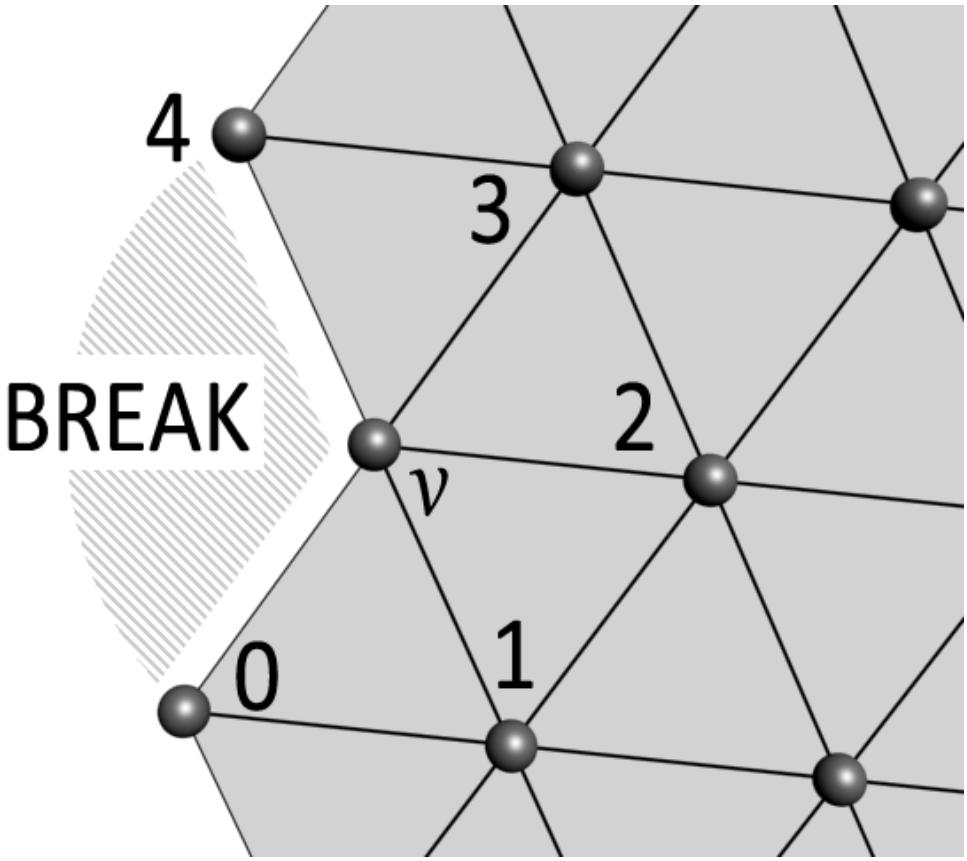


Figure 3.1: Example of a vertex at the open edge of a surface. The neighbor array of v would read $[0, 1, 2, 3, 4$,BREAK]. The BREAK flag instructs the simulation to use special routines for calculating curvature, area, etc. at v.

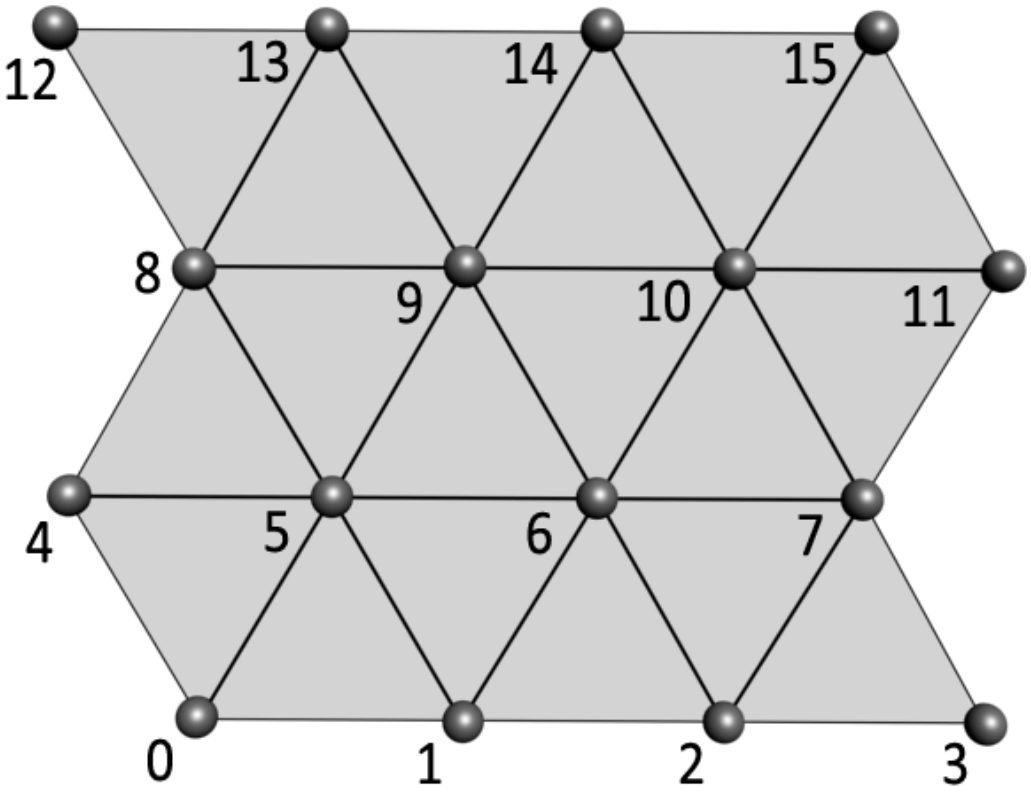


Figure 3.2: A 4×4 flat lattice constructed using equation 3.2. Indexing runs from bottom-left to top-right with offsets on alternating rows to produce a triangular lattice.

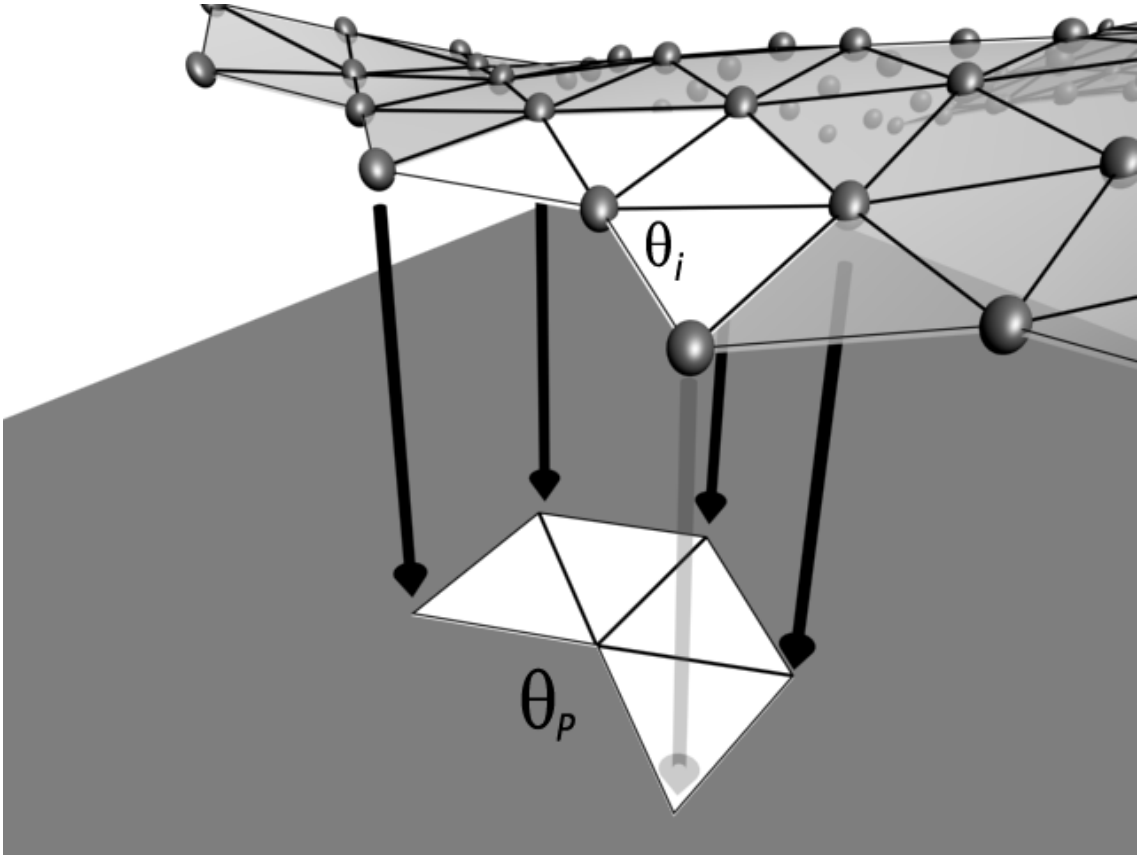


Figure 3.3: Calculating Gaussian curvature at the surface edge. The fan around a given vertex is projected onto the tangent plane. The angle subtended by the gap in the projection is measured as θ_P. The angle deficit is then computed using $2\pi - \theta_P$ as the expected angle for zero curvature (equation 3.3). The angle θ_i is the angle between adjacent edges around the vertex (not including the break).

$$G = \frac{(2\pi - \theta_P) - \sum \theta_i}{A} \tag{3.3}$$

This procedure satisfies the Gauss-Bonnet theorem (see supplement [1]) and produces a smoothly varying Gaussian curvature even at the edges of the surface.

3.3.4 Electrostatics Model

Electrostatic interactions from permanent molecular dipoles of lipid moieties and bound water might be interactions that compete with line tension [41]. In the context of large (micron-size) phase domains, the effects of electrostatics can be largely ignored due to electrostatic screening. The electric fields within the bilayer have a decay length on the order of a few nanometers, caused by the salt in the surrounding medium and the symmetric geometry of the bilayer [21, 10]. In the context of nanodomains, a few nanometers may represent several lipid shells, implying that electrostatic interactions cannot be ignored.

To model the electrostatics we define a dipole density at each vertex that depends only on the local phase,

$$\mu(v) = \begin{cases} \mu_d & \text{if } v \in L_d \\ \mu_o & \text{if } v \in L_o \end{cases} \tag{3.4}$$

The electric field at each vertex is defined by summing over the remaining vertices of the lattice and adding up the contribution of each according to the following [41, 25, 36],

$$\vec{E}(v) = \frac{1}{4\pi\epsilon} \sum_{x \neq v} e^{-k\|\vec{r}\|} \left[\frac{3(\vec{n} \cdot \vec{r})\vec{r}}{\|\vec{r}\|^5} - \frac{\vec{n}}{\|\vec{r}\|^3} \right] \mu(x) A(x), \tag{3.5}$$

where $\vec{n}$ is the membrane normal at x, $\vec{r}$ is the vector connecting v and x, k is the decay length of the fields within the membrane, and $A(x)$ is the area of vertex x. This equation has three important terms: the leading exponential function expresses the fact that the electric

fields within the plane of the bilayer decay with a characteristic length that depends on the salt concentration of the surrounding medium and the thickness and geometry of the bilayer; the portion in square brackets is the electric field contribution at v from a point dipole at position x (oriented normal to the membrane); the last term is the magnitude of this point dipole at position x. With this field defined, the energy functional becomes,

$$\mathcal{H} = \gamma L - \sum_{v} 2\mu(v)A(v)\vec{n} \cdot \vec{E}(v) \tag{3.6}$$

A factor of two is needed to take into account that both leaflets of the membrane experience a mirrored electric field. The assumed geometry of the lipid dipoles in the membrane and the field lines that are produced by equation 3.5 are shown in figure 3.4. Notice that the dipole density is always assumed parallel to the membrane normal. We make this assumption because whereas individual lipid dipole moments may orient away from the bilayer normal, the rotational symmetry of the liquid phase ensures that in-plane contributions to the dipole density will average to zero.

3.3.5 Quantifying Morphology: Radial Distribution Function

To study how the various parameters and interactions affect the clustering of phase domains, we use the $L_o - L_o$ radial distribution function, which gives the probability density of finding a vertex of L_o phase as a function of distance from another vertex of L_o. To estimate this distribution we produce a histogram with bin width $\langle \Delta r \rangle$, defined as the average distance between neighboring vertices. We define the set $B_v(i)$ as all vertices in a spherical shell around the vertex v[1],

$$B_v(i) = \{w \quad | \quad i \langle \Delta r \rangle < \|\vec{w} - \vec{v}\| < (i+1) \langle \Delta r \rangle\}, \tag{3.7}$$

[1]The symbol v refers to the scalar index of the vertex, while the symbol $\vec{v}$ refers to the vector position of the vertex in space.

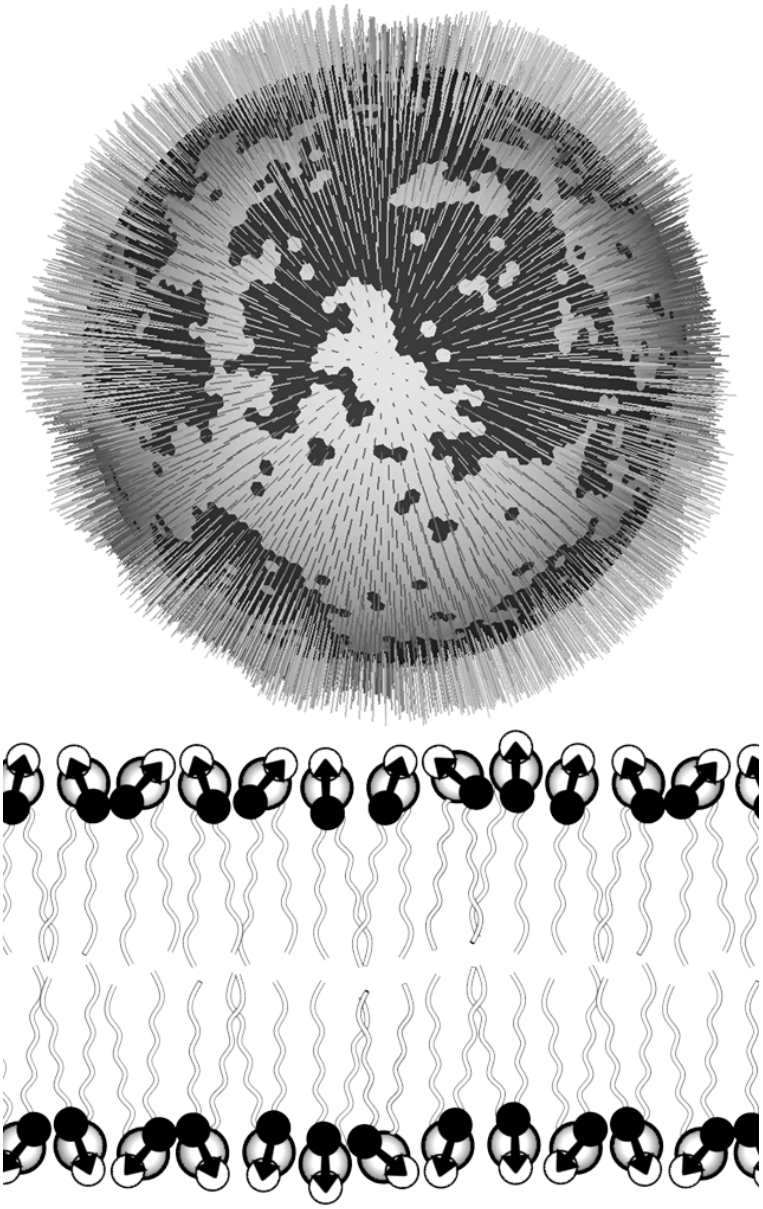


Figure 3.4: (Top) Electric field lines (as defined by equation 3.5) on the surface of a simulated GUV. (Bottom) Microscopic geometry of the charge distribution assumed in the formulation of the electrostatics model. Black beads represent point positive charges, white beads point negative charges, and black arrows the net individual dipole moments of each lipid.

and let $N_v(i)$ be the number of elements in $B_v(i)$. The $L_o - L_o$ correlation function can then be defined by counting the number of elements in each bin for each vertex and normalizing,

$$C(i\,\langle\Delta r\rangle) = \frac{1}{PN}\sum_{v\in L_o}\left[\frac{1}{N_v(i)}\sum_{w\in B_v(i)}\delta_{\phi_v\phi_w}\right] \tag{3.8}$$

where δ is the Kronecker delta function, P is the area fraction of L_o phase, and N is the total number of vertices on the lattice. In this treatment distance between vertices is measured along the contours of the surface and C is defined only at discrete intervals set by the bin width, $\langle\Delta r\rangle$.

The correlation function can be used to distinguish among the various morphology possibilities: random mixing, clustered domains, modulated phases, and macroscopic domains. These various morphologies are shown in figure 3.5. Random mixing yields a constant value of $C(r) = P$ (where P is the area fraction of L_o phase); clustered domains have an exponential decay that defines a correlation length; modulated phases show a distinct periodicity that defines the repeat length of the patterns; and macroscopic domains yield a roughly linear or sigmoidal correlation function that crosses the line $C(r) = P$ at the domain size.

3.3.6 Heat Capacity

Heat capacity is useful to assess how competing interactions change the thermodynamic properties of the model. The heat capacity at constant volume can be defined as,

$$h \propto \left\langle \mathcal{H}^2 \right\rangle - \langle \mathcal{H} \rangle^2 \tag{3.9}$$

It is sufficient for our purposes to define a quantity that is proportional to the actual heat capacity because we are interested in the *locations* of features of the curve, such as peaks or discontinuities, rather than numerical values. Because heat capacity is a statistical measure,

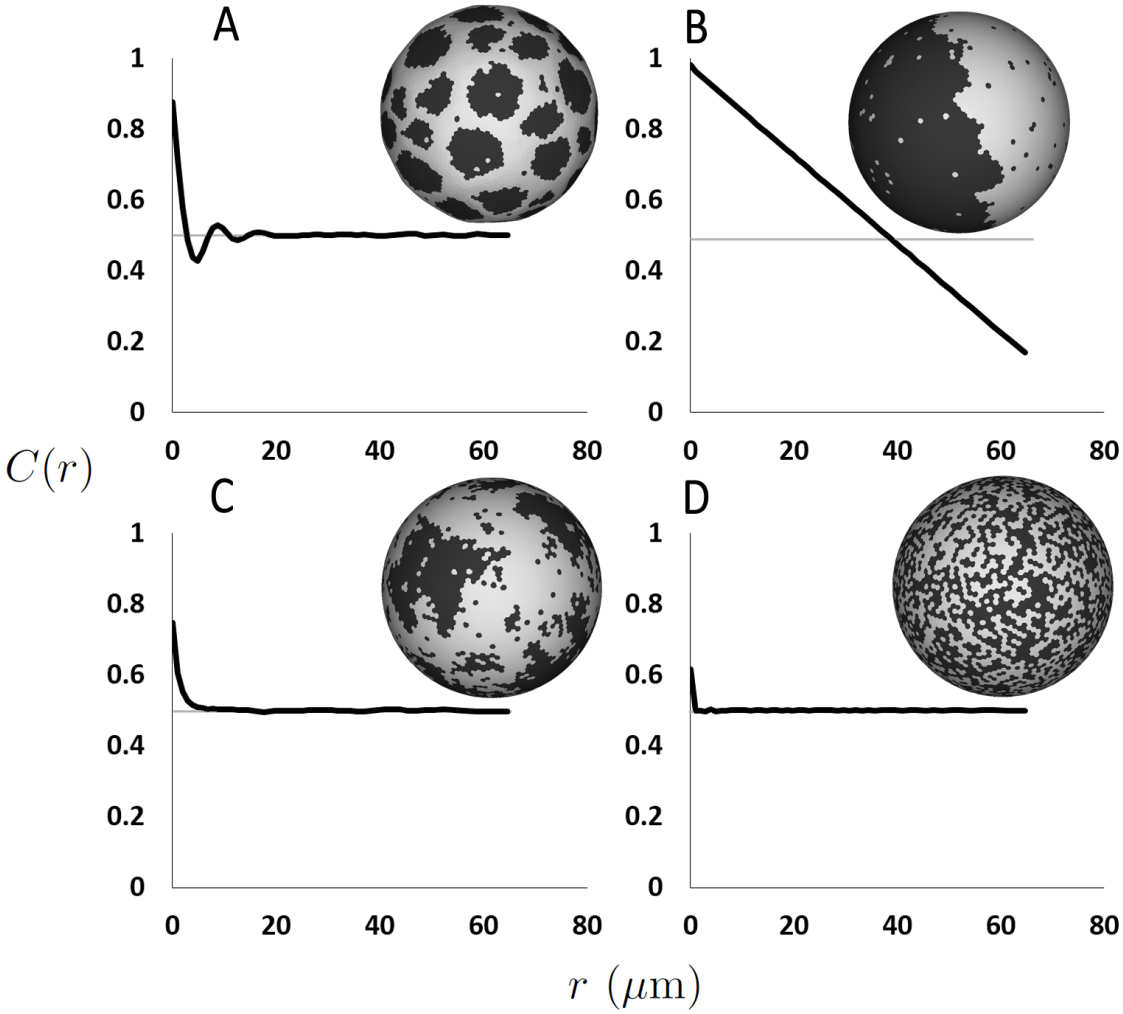


Figure 3.5: Different morphologies give rise to distinct shapes of their correlation function, which measures the probability of finding like-phases at a distance r from each other. Modulated phases (A) show damped oscillations corresponding to the repeat length of the patterns. Macroscopic domains (B) show a linear or sigmoidal decrease (see figure 3.10) that crosses the $C(r) = P$ axis at approximately the domain size. Clusters and critical fluctuations (C) show an exponential decay that defines a correlation length; and random mixing (D) yields a straight horizontal line at $C(r) = P$. In the above examples, area fraction $P = 0.5$.

we must consider error and correlation in the measurements. Heat capacity was measured by taking 1000 samples of the total energy at intervals separated by $100 \times N$ Monte Carlo steps (where N is the number of vertices on the lattice) to minimize autocorrelation. This process still produces curves with high variance, but the important features are readily visible.

3.4 Results and Discussion

3.4.1 Line Tension and Renormalization

One useful feature of this coarse-grained simulation approach is its applicability to a wide range of size scales. By changing the size and resolution of the lattice one can simulate from 100μm diameter GUVs, where each vertex represents hundreds of thousands of lipids, down to 0.06μm diameter LUVs, where each vertex represents only a few lipids. In order to relate the simulation results to physical observations it is important to consider the problem of running coupling constants [6]. This is the tendency of certain parameters to change when degrees of freedom are lost in coarse graining.

This problem becomes apparent when comparing simulations with the same parameter set and changing only the resolution of the lattice. Parameters that give macroscopic phase separation at lower resolution (10,000 vertices) can yield random mixing on a higher resolution lattice (40,000 vertices). This is due to the line tension being a running coupling constant. In this section we consider a Hamiltonian that includes only the contribution from line tension, $\mathcal{H} = \gamma L$, in order to determine how it changes under coarse-graining. Bending modulus can be treated in a similar way [20]. To compare line tension on different resolution lattices we define the scale invariant quantity $u \equiv \gamma l_0$, where l_0 is the average boundary length contribution for a pair of adjacent vertices. This makes our model roughly equivalent to an edge-counting Ising model,

$$\mathcal{H} = \sum_{\langle i,j \rangle} \frac{\gamma l_0}{2}(1 - S_i S_j) \tag{3.10}$$

$$= \sum_{\langle i,j \rangle} -\frac{u}{2} S_i S_j, \tag{3.11}$$

where $S_x = (2\phi_x - 1)$ and we have discarded the constant energy offset. In this form we can relate our simulation to the known solution of the Ising model on a triangular lattice [15]. Criticality for this system occurs when,

$$\frac{u_{\text{crit}}}{kT} = \frac{\ln(3)}{2} \approx 0.55. \tag{3.12}$$

For u/kT below this critical value the lattice simulation should function exactly as a triangular lattice Ising model with zero external field. We can test this by measuring the heat capacity as a function of u/kT, shown in figure 3.6. The heat capacity curves, $h(u/kT)$, for both the flat and spherical lattice show no remarkable differences and most importantly both show a sharp peak at $u/kT \approx 0.55$, consistent with the theoretical value for the critical point of the triangular lattice Ising model. This close match in critical point is surprising because the flat lattice has a valence of exactly 6, whereas the spherical lattice has many 5-7 defects and only an average valence of 6.

The next step is to formulate a means of relating the morphology on lattices at different grain levels. The grain level is defined as the number of lipids represented by each vertex, $G \equiv a_v/a_0$, where a_0 is the area of a single lipid and a_v is the average area of a single vertex. Two morphologies at different grain levels are defined to be equivalent if both have the same inverse correlation length, where the inverse correlation length is given by,

$$c(u/kT) \equiv \left(- \left. \frac{dC(r)}{dr} \right|_{r=0} \right) - c_{\text{crit}}. \tag{3.13}$$

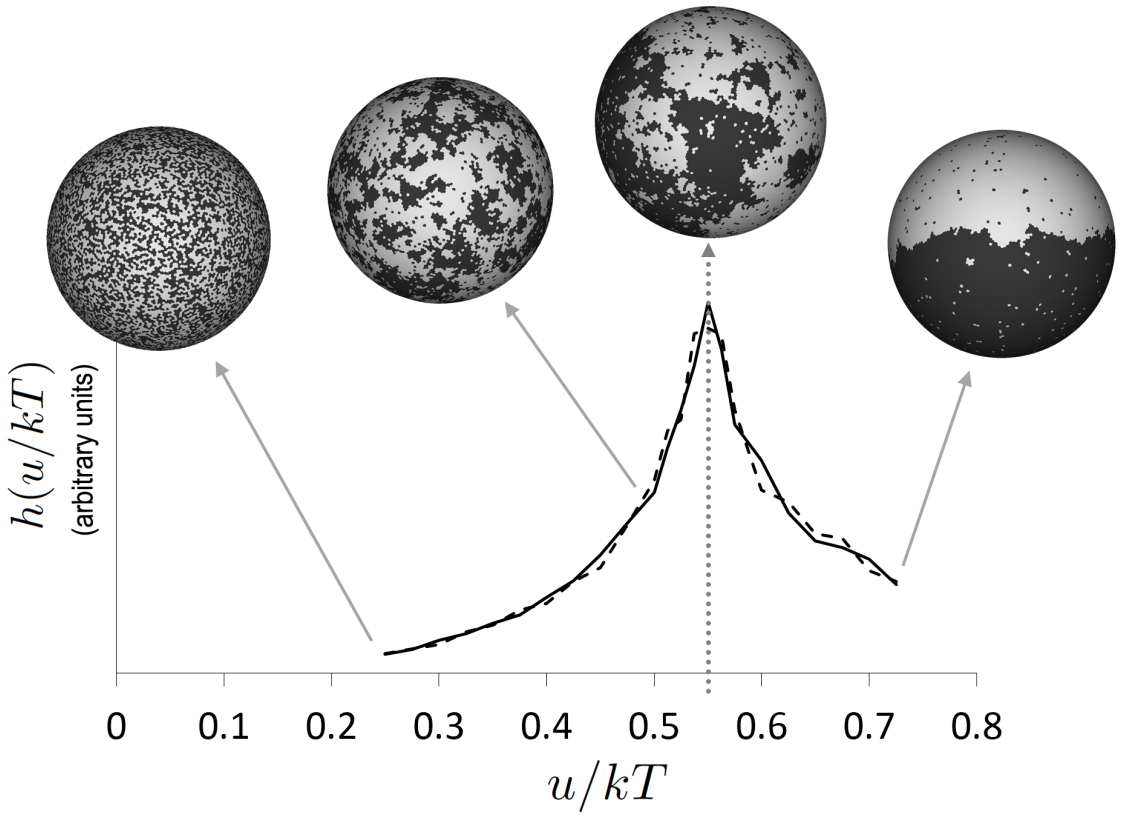


Figure 3.6: Heat capacity as a function of line tension for both spherical (solid black) and flat (dashed black) lattices. The peak corresponds to $u/kT = 0.55 \approx \ln(3)/2$, the theoretical critical point of the triangular Ising model.

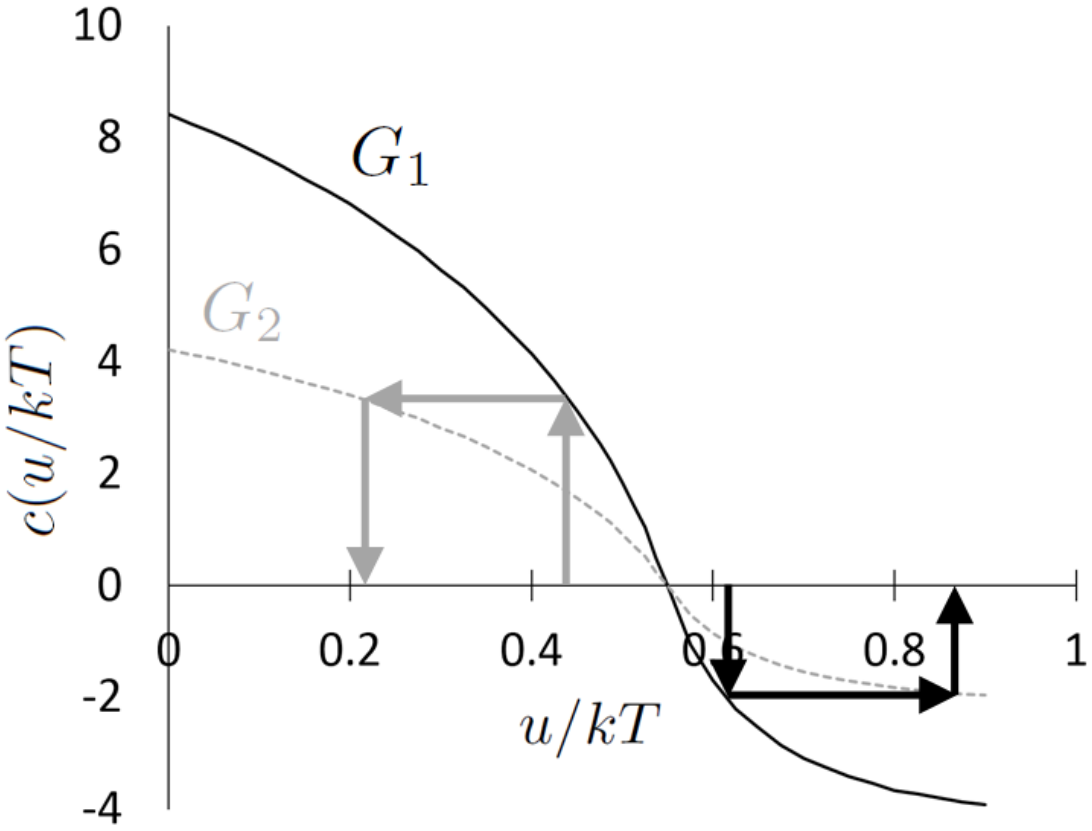


Figure 3.7: Inverse correlation length relates line tension at two different grain levels, G_1 (solid black) and G_2 (dashed grey). A value of u_1/kT is chosen for grain level G_1. The corresponding value for G_2 is determined and the value of u_2/kT is read off the axis. Values of u/kT below the critical point shift towards zero under coarse-graining (grey arrows), whereas values above the critical point shift towards ∞ (black arrows).

Shifting this curve by the value of c_{crit} fixes the inverse correlation length at the critical point to be zero ($c(u_{\text{crit}}/kT) = 0$), thus ensuring that the critical point is a fixed point under renormalization. The value c_{crit} is determined empirically through simulation.

The curves for $c(u/kT)$ at two different grain levels and an illustration of the renormalization procedure are shown in figure 3.7. Notice that $c(u/kT)$ need only be measured at a single grain level G_1, because changing the grain level to G_2 only introduces a constant prefactor of $\sqrt{G_1/G_2}$. When coarse-graining from grain level G_1 to G_2, the condition that the inverse correlation length remains fixed is given by the scaling relation,

$$\frac{c(u_2/kT)}{\sqrt{G_2}} = \frac{c(u_1/kT)}{\sqrt{G_1}} \tag{3.14}$$

For any given values of u_1,G_1, and G_2 this relation is numerically solved for u_2, giving rise to a set of renormalization flows as grain level G_2 increases. The flows are shown in figure 3.8. By following these flows we can relate the value of u from one grain level to another, and hence make meaningful comparison of the resultant phase morphologies when other parameters (such as vesicle size) are varied.

The renormalization flows show several important aspects of the scaling behavior in the simulation. Below the critical point, u tends towards 0, indicating clusters of finite size, which will eventually be below the lattice resolution. At the critical point (black dashed line) renormalization does not change line tension because the morphology is fluctuating on all length scales in a self-similar way [14, 26]. Above the critical line tension, u tends towards ∞. This is because under-resolving the boundary of a large domain is equivalent to damping out its fluctuations, which shows up as an increase in line tension.

Figure 3.9 shows how this scaling procedure affects the morphology of lattices at different grain levels. As an example we illustrate a lattice of $40,000$ vertices being coarse-grained to a $10,000$ vertex lattice (holding vesicle size fixed), giving a grain level of $G_2/G_1 = 4$. For naive coarse-graining, where line tension is treated as a true energy per unit length, the morphologies can be radically different and comparison becomes meaningless. Using the scale-invariant form of line tension, u, for both grain levels, $u_1/kT = u_2/kT = 0.5$, there is qualitative agreement between the morphologies, but the linear dimension of the structures is still off by a factor of $\sqrt{G_2/G_1}$. Using renormalization flows for $u_1/kT = 0.5$ and $G_2/G_1 = 4$, the value $u_2/kT = 0.425$ is used for the coarse-grained lattice (solid black line in figure 3.8). These values produce both qualitative agreement and structures of the same linear dimension.

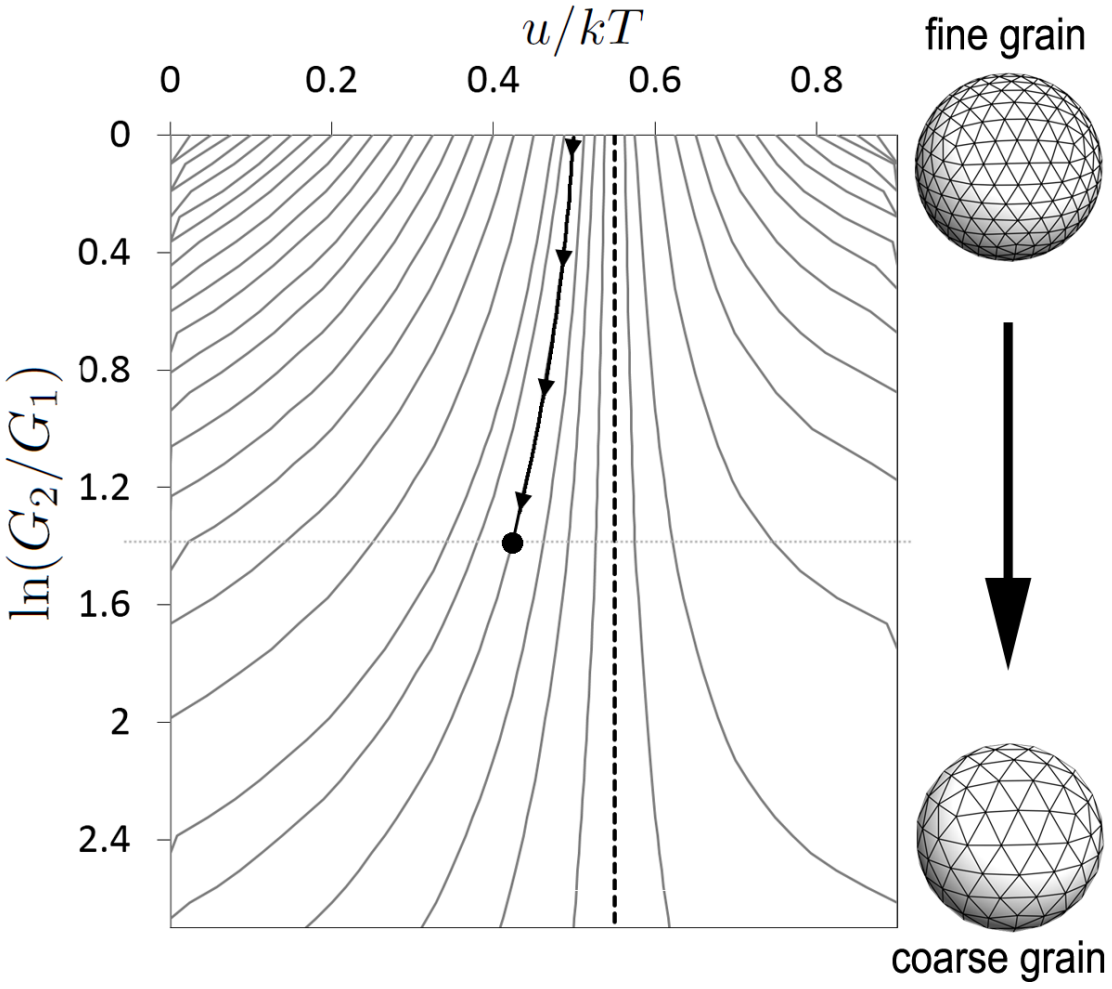


Figure 3.8: Renormalization flows of u/kT going from a fine grained simulation (top) to a coarse grained simulation (bottom). To relate u at two different grain levels, follow the contours from the top down to the appropriate value of $\ln(G_2/G_1)$. For example, coarse graining by a factor of $G_2/G_1 = 4$ ($\ln(G_2/G_1) = 1.39$, dotted grey line) with a line tension of $u_1/kT = 0.5$ (black line with arrows) gives a renormalized value of $u_2/kT = 0.425$. The critical manifold (dashed black line) remains fixed under renormalization.

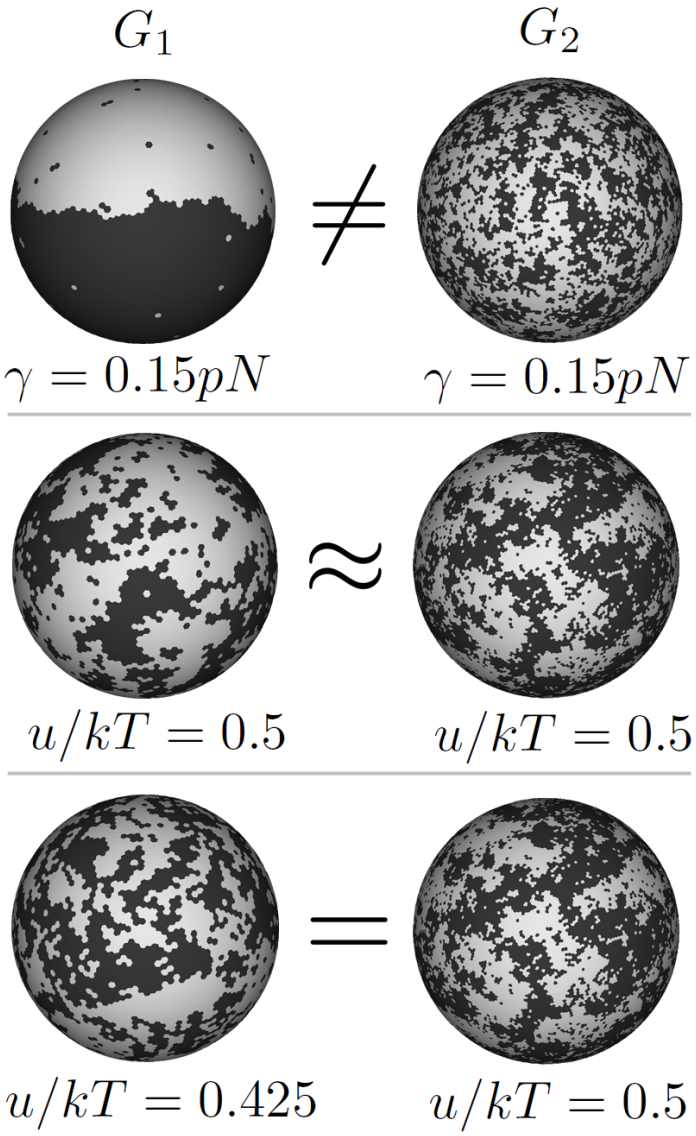


Figure 3.9: Renormalization is required in order to compare simulations having different levels of coarse-graining. The two lattices above have $10,000$ vertices (left G_2) and $40,000$ vertices (right G_1). Holding the true line tension fixed under coarse graining (top) leads to radically different morphologies. Holding the scale-less line tension fixed (middle) at $u_1/kT = u_2/kT = 0.5$ produces qualitatively similar morphology, but different correlation length. Using the renormalization flows (bottom), coarse-graining by $G_2/G_1 = 4$ yields $u_2/kT = 0.425$. This produces equivalent morphologies with similar correlation length.

parameter	value	unit
u/kT	0.7	N/A
κ_d,$-\bar{\kappa}_d$	10×10^{-19}	J
κ_o,$-\bar{\kappa}_o$	80×10^{-19}	J
P	0.5	N/A
R	25	μm
kT	4.0×10^{-21}	J

Table 3.1: Default parameter set for studying effects of background curvature on modulated phase patterns.

This exercise in renormalization outlines the potential pitfalls of coarse graining a model when parameters that are sensitive to the removal of degrees of freedom may be present. We have also outlined a general procedure for empirically solving renormalization flows whenever an equivalence class can be defined that is independent of the grain level (in our case, phase morphologies with the same inverse correlation length).

3.4.2 Flat and Spherical Lattices

Previous simulations have shown that the competition between line tension and curvature energies, defined by the Helfrich energy functional (equation 3.1), can stabilize modulated phases. These results closely match the characteristics of modulated phase patterns observed on GUVs in the four component lipid system DSPC/DOPC/POPC/chol. All of the previous simulations were performed on a triangulated spherical surface to best approximate the shape and size of the observed GUVs. The spherical lattice imposes a constant background curvature that breaks the symmetry of the energy functional and may play an important role in the formation of modulated phase patterns. To better understand the role that this symmetry-breaking plays in the formation of modulated phases we perform simulations on spherical and flat lattices to compare the resultant phase morphology.

Modulated Phases Do Not Appear on Flat Lattices for Parameters That Produce Modulated Phases on Spherical Vesicles

Simulations were performed with the parameters shown in table 3.1. The value of line tension, $u/kT = 0.7$ ensures that the model is in the two-phase region, but is low enough so as not to dominate the energy landscape. The bending moduli are slightly higher than literature reported values [27], but are chosen to produce modulated phase patterns. Setting $\bar{\kappa} = -\kappa$ is consistent with literature estimates [17]. These parameters give modulated phases when used on a 50μm diameter spherical surface. When these same parameters are used on a flat lattice with dimension $96 \times 83\mu\text{m}^2$ (equal area and vertex density), modulated phases do not appear. Simulation correlation functions and the representative morphologies are shown in figure 3.10. This finding implicates background curvature as a stabilizing factor in curvature-induced modulated phases.

Background Curvature is Important in Stabilizing Modulated Phases

To study how background curvature affects the formation and stability of modulated phases a series of simulations was performed in which only the background radius of curvature was changed. To produce different background curvatures the spherical and flat lattices were molded, scaled, and truncated to produce open spherical caps with the same vertex density. The observed changes in morphology are shown in figure 3.11. We find that for the parameter set explored, background curvature is necessary for the stability of patterns. Increasing the radius of curvature by about 50% (to 35μm) causes a broadening of the domain size while a doubling of the radius of curvature (50μm) is sufficient to altogether arrest the formation of a well-defined periodic pattern. Macroscopic domains persist as the radius of curvature is further increased. The sudden onset of modulated phases as the surface is curved represents a buckling instability that exchanges the energy cost of bending a large L_o domain for

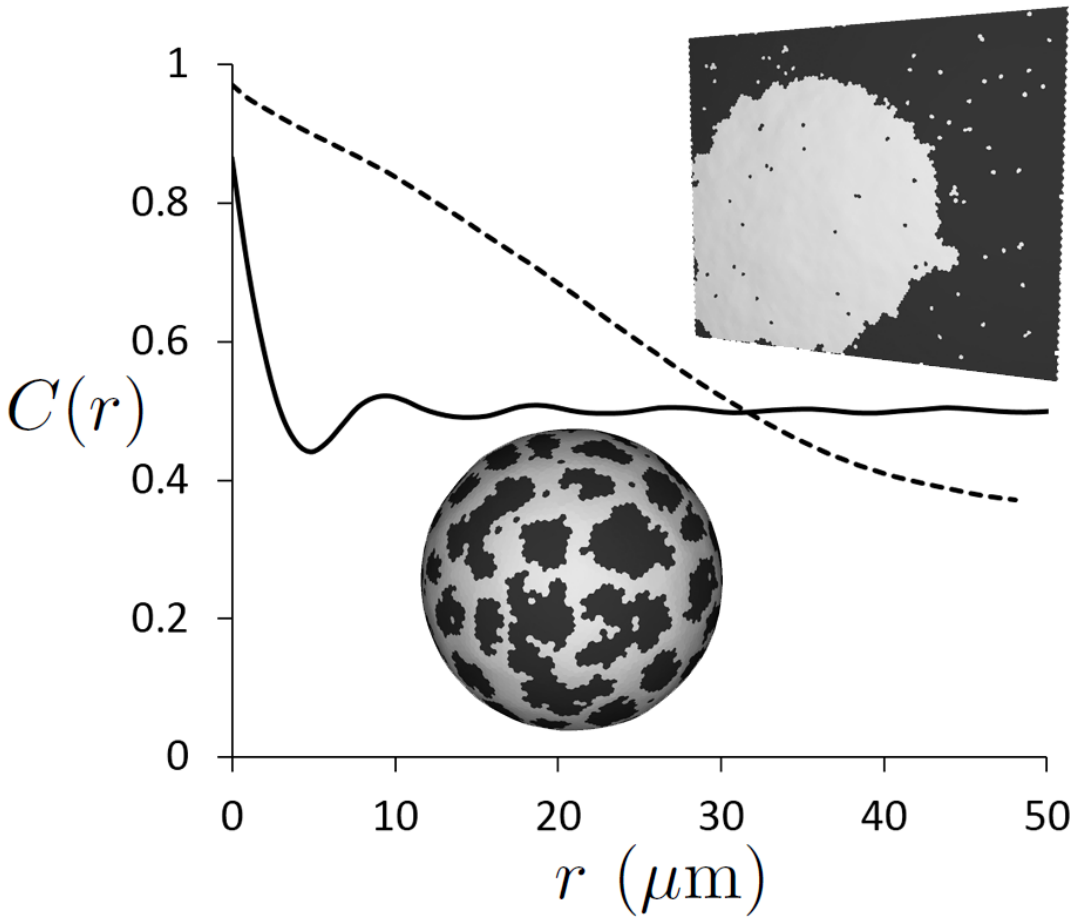


Figure 3.10: With parameters that give rise to modulated phases (table 3.1) on a spherical lattice (solid curve), modulated phases do not appear on a flat lattice (dashed curve).

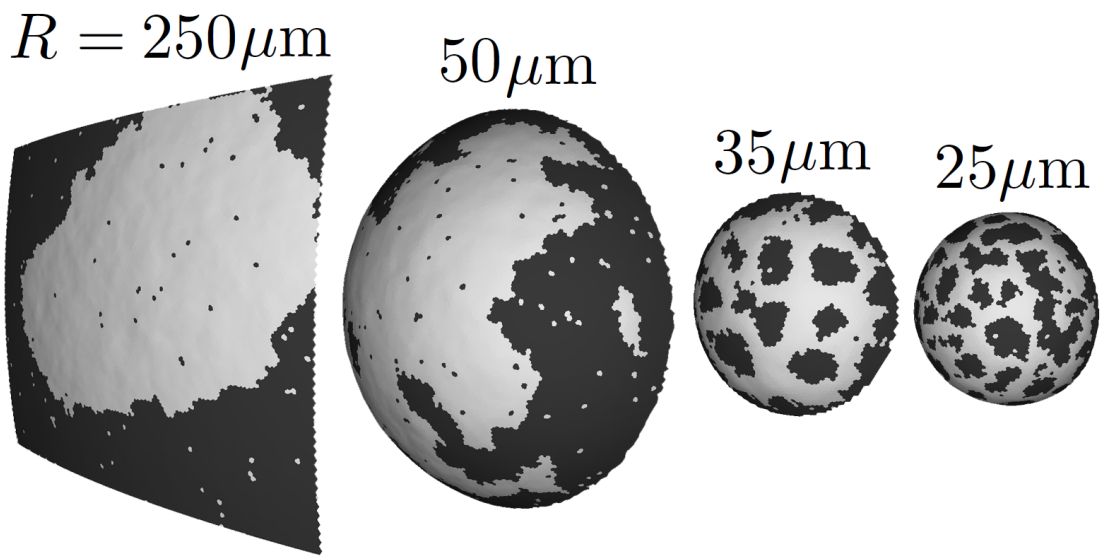


Figure 3.11: Background curvature is important to stabilizing modulated phases. From right to left the background radius of curvature is increased. Each simulation shown has the same area per vertex (0.78μm^2/vertex) and is drawn on the same absolute length scale for comparison.

an increase in phase boundary. Similar curvature-induced morphological instabilities have been observed that minimize the phase boundary using out of plane deformations of the membrane [42]. These observations may have implications for raft morphology, given the large variations in curvature observed for living cells [30].

3.4.3 Nanodomains and Electrostatics

So far the simulation outlined successfully models GUVs with optically resolvable modulated phases and macroscopic domains. To study the nanoscopic phase separation observed in some model systems [8, 18, 40, 43], the simulated vesicle is scaled down so that each vertex represents only a handful of lipids. At this scale nanometer-size domains are resolvable, allowing us to explore their morphology and stability. Here we simulate vesicles that are 60nm in diameter with a resolution of 10,000 vertices, giving a grain level of approximately 2 lipids per vertex. Our motivation for this particular vesicle size is that 60nm vesicles are

used by Heberle et al. in small angle neutron scattering to study the size and distribution of nanodomains [12].

When each vertex represents only a few lipids, it is necessary to consider electrostatic dipole interactions, which vary on this length scale. Previous calculations of the range of the electric fields within bilayers showed a decay length between 2-4nm depending on the concentration of salt in the surrounding medium [21]. For GUV simulations of optical phenomena this allowed treating dipole interactions as a contribution to the mechanical parameters describing the Helfrich energy functional, and enabled obtaining bending modulus values directly from experiment [27]. At the nanodomain scale this decay length represents several lipid shells and thus must be treated as a separate term in the energy functional as described by equations 3.5 and 3.6. Since the microscopic origin of the electrostatic interaction is approximated as adjacent, parallel molecular dipoles, it is necessarily a dispersive interaction that will compete with line tension. This provides a potential mechanism for the stabilization of nanoscopic phase separation.

Electrostatic Parameters

To implement this model we start with approximate values for the electrostatic parameters. The key parameter that controls the magnitude of the interaction is the dipole density of each phase, μ_d and μ_o. The most robust measurement of this parameter comes in the form of the dipole potential Ψ_d [44], which is the difference in electrostatic potential between the surface and interior of the membrane. This value is on the order of 300mV for the L_d phase and has been estimated to be as high as 1000mV for the L_o phase [45, 39]. In order to convert such a measurement to dipole density we must assume a geometry for the charge distribution. The simplest geometry that captures the essential features of the transmembrane potential has the headgroups as a parallel plate capacitor. A positively charged plate is placed at the headgroup/chain interface and a negatively charged plate at

the headgroup/water interface [41, 24]. This provides the simple relation $\mu = \epsilon\Psi_d$ for the dipole density, where ϵ is the effective dielectric experienced by the molecular dipoles.

The dielectric constant ϵ for our simulation is not well-determined. The electrostatic environment within the headgroup region of the membrane is very complicated. It has a high degree of anisotropy with estimates of the normal and lateral components of the dielectric tensor ranging over two orders of magnitude [29, 33]. We constrain our value of ϵ by noting that the major contributions to the dipole density come from the carbonyl groups that link the acyl chains to the glycerol backbone, and from bound water [31]. The zwitterionic phosphotidylcholine headgroup is oriented nearly parallel to the membrane surface and thus contributes little to the normal component of the dipole density [39]. With this in mind we choose $\epsilon = 8$ to emphasize that the carbonyls are partially submerged in the hydrocarbon portion of the bilayer, which has a low dielectric [44, 29, 24, 5]. Using these values in the parallel plate capacitor equation yields an estimate for the L_d dipole density of $\mu_d = 133$ e^-/μm. For the L_o phase we use the value $\Psi_d = 700$mV [45], which gives a dipole density of $\mu_o = 309$ e^-/μm. This contrast between the two phases, $\Delta\mu = (\mu_o - \mu_d) = 176$ e^-/μm, produces the frustration that could drive the break-up of macroscopic domains.

The other parameter in equation 3.5 that needs to be addressed is the decay length of the electric fields within the plane of the bilayer, k^{-1} (not to be confused with the Debye length in the aqueous surrounding medium). This parameter is well constrained and depends on the salt concentrations in the surrounding medium and the geometry of the bilayer. Previous simulations of the Debye-Huckel equation allowed solving for the electrostatic potential in and around a circular phase domain and enabled calculating this decay length [21]. See Supplemental Material for a detailed explanation of these calculations. We found the accessible range of decay lengths from $k^{-1} \approx 1$nm at arbitrarily high salt concentrations up to $k^{-1} \approx 4$nm for a surrounding medium of pure water. We used a default value $k^{-1} = 2$nm, corresponding to a physiological concentration of salt in the surrounding medium.

parameter	value	unit
u/kT	0.7	N/A
μ_d	133	e^-/μm
μ_o	309	e^-/μm
k^{-1}	2	nm
κ_d,$-\bar{\kappa}_d$	2×10^{-19}	J
κ_o,$-\bar{\kappa}_o$	$\{8, 14, 20\} \times 10^{-19}$	J
R	30	nm
P	0.5	N/A

Table 3.2: Default parameter set for studying electrostatic effects on morphology.

In sections III.C.2-4 we consider only the competition of line tension and electrostatics as given by equation 3.6. In section III.C.5 we consider the added complexity of curvature in tandem with line tension and electrostatics. Line tension is chosen to be $u/kT = 0.7$ because this yields the two-phase region while still exhibiting interesting behavior with the electrostatic parameters derived above. We set the phase fraction of the membrane to be $P = 0.5$ to emulate being in the center of a tie-line far from any phase boundaries in composition space. The default parameter set for our exploration of electrostatics is shown in Table 3.2 unless otherwise stated. The bending moduli shown only apply to section III.C.5 where Helfrich curvature terms are included.

Contrast in Dipole Density Between Phases Breaks Up Phase Domains

The contrast in dipole density between the two phases, $\Delta\mu$, dictates the magnitude of the electric fields. To explore how the dipole interactions disrupt the phase domain morphology we examine the correlation functions for a series of simulations with $\Delta\mu = 0$ up to $\Delta\mu = 176$. To achieve this contrast the value of μ_o is changed while the value of μ_d is held fixed at the value given in table 3.2. The results of this series are summarized in figure 3.12.

For $\Delta\mu = 0$ the electrostatic interactions are irrelevant to the energetics. A single hemispherical domain exists, as indicated by the linear correlation function crossing the line

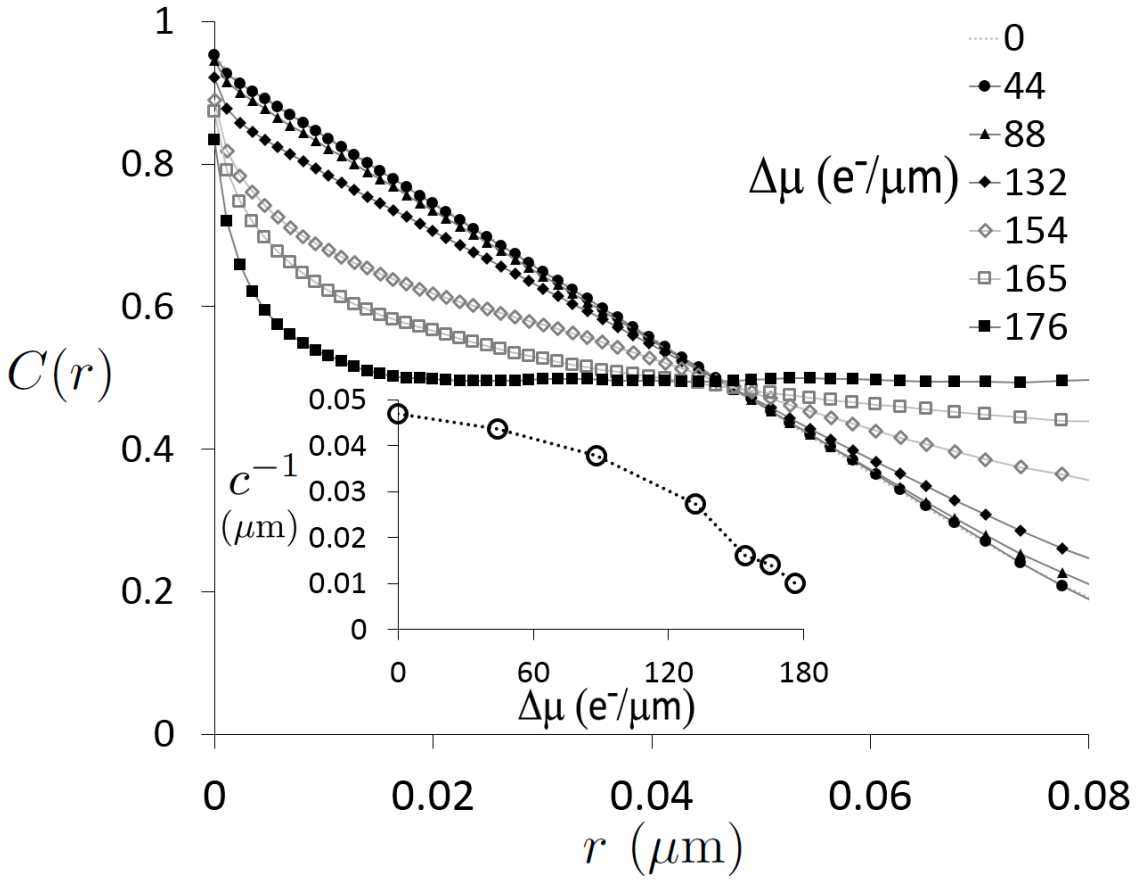


Figure 3.12: Correlation functions for a series of dipole density contrasts $\Delta\mu$. As dipole contrast is increased the macroscopic domain morphology (straight line) abruptly transitions to an exponential decay. (Inset) Correlation length as a function of $\Delta\mu$.

$C(r) = P$ at $r \approx 45$nm. As $\Delta\mu$ increases, the correlation function abruptly shifts from linear to an exponential decay curve. This indicates that the hemispherical domain has dispersed into smaller irregular clusters. For dipole density contrast greater than $\Delta\mu = 130\ e^-/\mu$m correlation lengths range from 10-20nm (inset of figure 3.12), consistent with the estimated size of nanodomains.

Inclusion of Dipoles Shifts Critical Line Tension to a Higher Value

To better understand the mechanism behind the transition from macroscopic to nanoscopic phase separation caused by dipolar repulsion, we measure how the miscibility critical point has shifted. The heat capacity is measured as a function of u/kT for a dipole density contrast of $\Delta\mu = 176\ e^-/\mu$m and decay length of $k^{-1} = 2$nm. The results of this calculation are shown in figure 3.13.

The inclusion of a dipolar repulsion term shifts the miscibility critical point to a higher value of u/kT than that of the triangular lattice Ising model. The critical point now occurs at $u_{\text{crit}}/kT \approx 0.72$. Once the dipole density contrast is high enough to shift the critical point above the current line tension ($u/kT = 0.7$) the system transitions from two coexisting phases to one phase. This shows why the transition observed when $\Delta\mu$ is increased occurs so abruptly. This finding is interesting; it shows a sharp transition from macroscopic to nanoscopic phase separation that is sensitive to a specific aspect of membrane composition. The local composition of the plasma membrane is known to be under tight regulation by the cell, thus providing a possible mechanism for the formation and dissipation of membrane rafts.

Decay length of Electric Fields Influences Domain Size

The other parameter important for describing dipolar repulsion in the membrane is the decay length of the fields within the plane of the bilayer, k^{-1}. Whereas the contrast $\Delta\mu$ describes

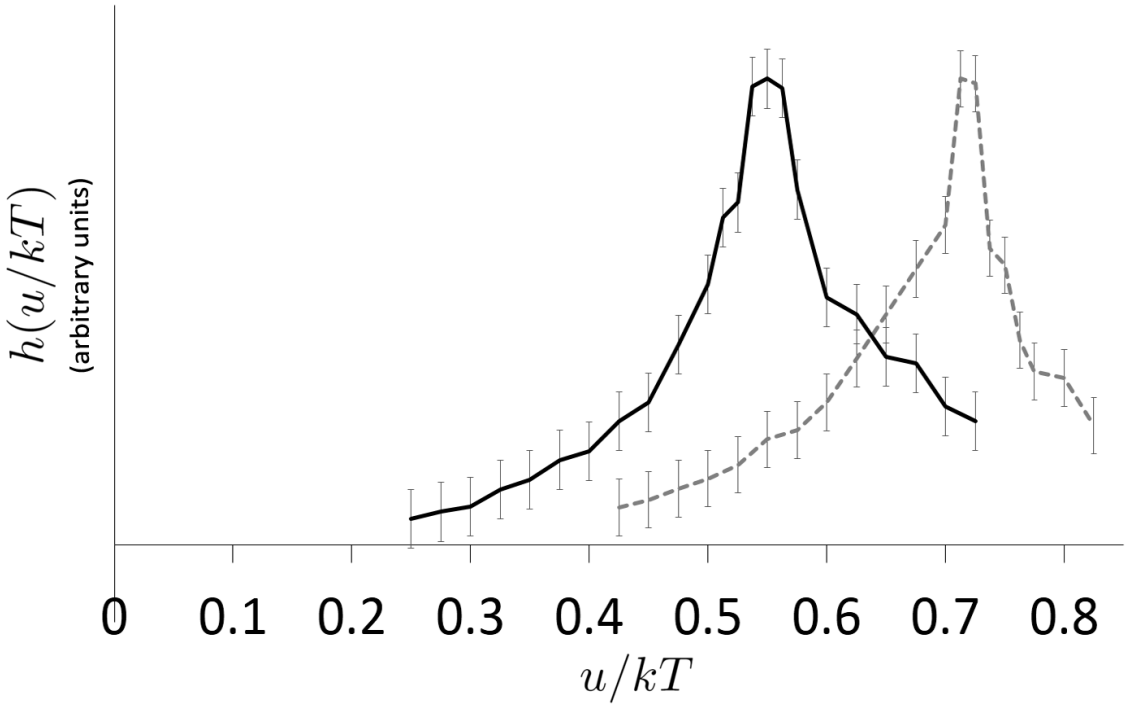


Figure 3.13: Heat capacity for a spherical lattice without (black) and with electrostatic repulsion (grey dashed, parameters in Table I). Electrostatics shifts the critical point to the right, explaining why increasing dipole contrast can lead to an abrupt shift between macro and nanodomains.

the magnitude of the electric fields, the decay length describes how many shells of lipid are affected by the dipole moment of a given lipid. With the geometry of the bilayer fixed, the only external factor governing the decay length is the concentration and type of ions in the surrounding medium. For physiological conditions this decay length is $1.5 - 2$nm (see supplement [1]). To explore how the decay length (and by extension the salt concentration) affect the phase morphology we performed a series of simulations, varying the decay length from $k^{-1} = 1$nm (high salt concentration) up to $k^{-1} = 3$nm (low salt concentration). The resultant correlations functions are summarized in figure 3.14.

We find that as the decay length is increased the phase morphology shifts from macroscopic domains to clusters with progressively shorter correlation length (inset of figure 3.14). For the shortest decay length, $k^{-1} = 1$nm, the electric fields only extend out roughly two lipid shells. Such a short decay length limits dipolar repulsion to a local effect that does not disrupt large scale phase separation, as indicated by the linear correlation function.

As the decay length is extended, dipolar repulsion affects more and more shells of lipid. The overlap in the electric fields from nearby lipids becomes more significant and frustrates the formation of a single large domain. This decrease in stability becomes apparent as the correlation function peels away from the macroscopic case and take on the characteristic shape for irregular clusters.

The range of values examined here effectively covers the range of physically realizable decay lengths. The shorter decay lengths of $1-2$nm are expected for physiological conditions on the cytosolic side of the plasma membrane. The larger decay lengths of ≥ 3nm are typical of conditions in GUV experiments, where sucrose/glucose solutions are used to provide density and optical contrast.

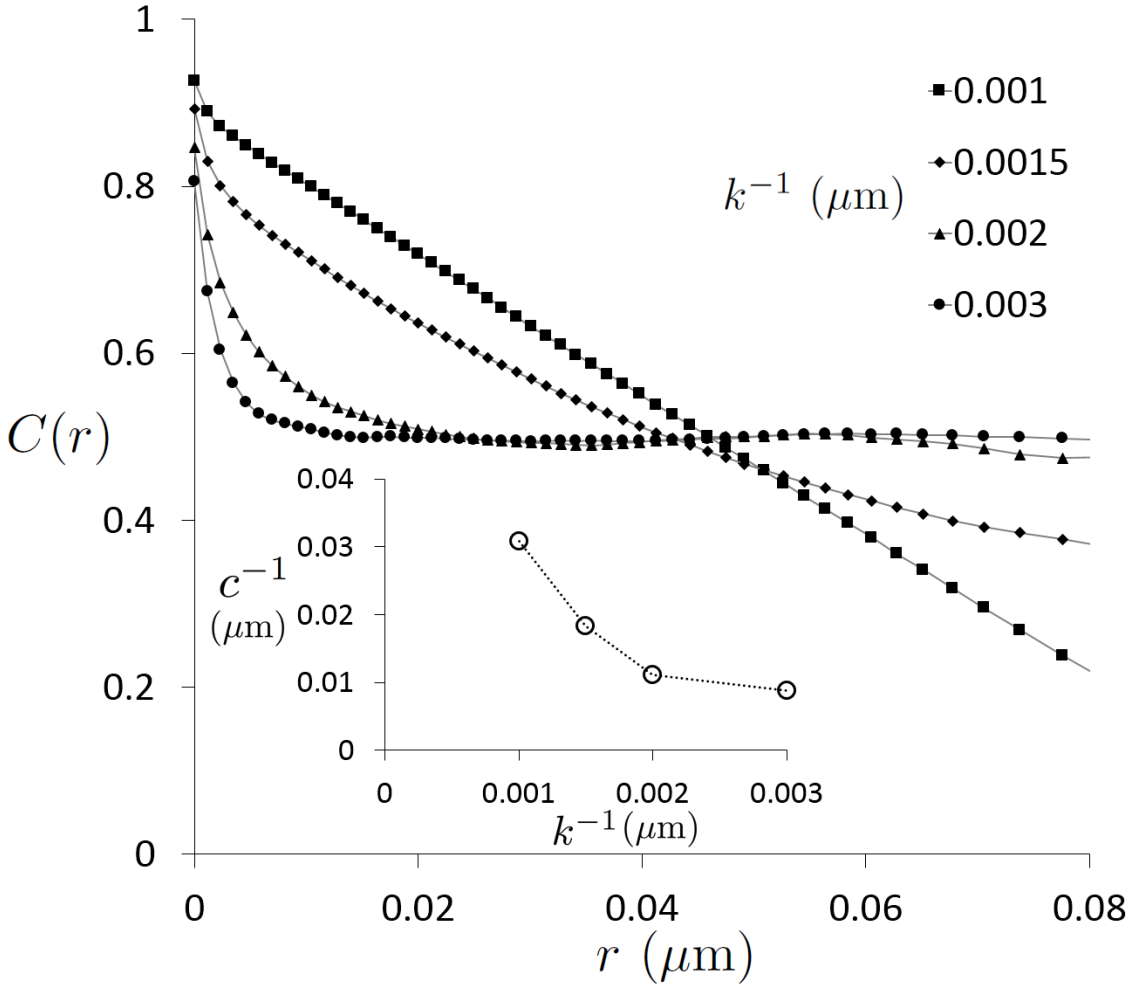


Figure 3.14: Correlation functions for a series of decay lengths from $k^{-1} = 1$ nm to $k^{-1} = 3$ nm. The smallest decay length leads to macroscopic phase separation, which transitions smoothly to small clusters as k^{-1} is increased. (Inset) Correlation length as a function of k^{-1}, shows that for $k^{-1} > 1.5$nm, correlation lengths are on the order of 10 nm.

Curvature and Electrostatics Stabilize Circular Domains on LUVs

In our previous study of competing interactions and phase morphology [2], curvature was used as a competing interaction with line tension to break up macroscopic domains and stabilize modulated phases at an optical scale. Here we have shown that electrostatic repulsion is another interaction that can compete with line tension to produce phase morphology at nanodomain size scales. The next step in complexity for this model is to consider both electrostatics and curvature terms working in tandem and assess how the phase morphology is affected.

The simulations in this section use the energy functional defined by equation 3.6 but also include the Helfrich curvature energy terms from equation 3.1. As a starting point for our exploration, bending modulus values are chosen similar to those measured by Semrau et al. [34] ($\kappa_d = 2 \times 10^{-19}$J, $\kappa_o = [8 - 20] \times 10^{-19}$J) and the remaining parameters from table 3.2. A series of simulations were performed for increasing value of κ_o (and by extension $\bar{\kappa}_o$). The results of these simulations are shown in figure 3.15.

When electrostatic terms are not included these curvature parameters alone were not sufficient to disperse a macroscopic domain (not shown). By contrast, electrostatics alone (E) strongly opposed the formation of a single macroscopic domain, producing a correlation length of about 9nm. When curvature is included together with electrostatics (C(κ_d, κ_o)+E), the simulated domains become *larger*, e.g. the correlation length increases from 9nm to 12nm. This surprising result shows that curvature is not simply an interaction that opposes line tension, but instead works in a direction to minimize the frustration of bending the membrane. As κ_o is increased further, *modulated* phases precipitate with a *smaller* characteristic size of approximately 8nm in diameter. Experimentally measured bending moduli already contain contributions from the dipole repulsion of nearby lipids, so we must be mindful of this when interpreting the values used to produce modulated phases in figure 3.15.

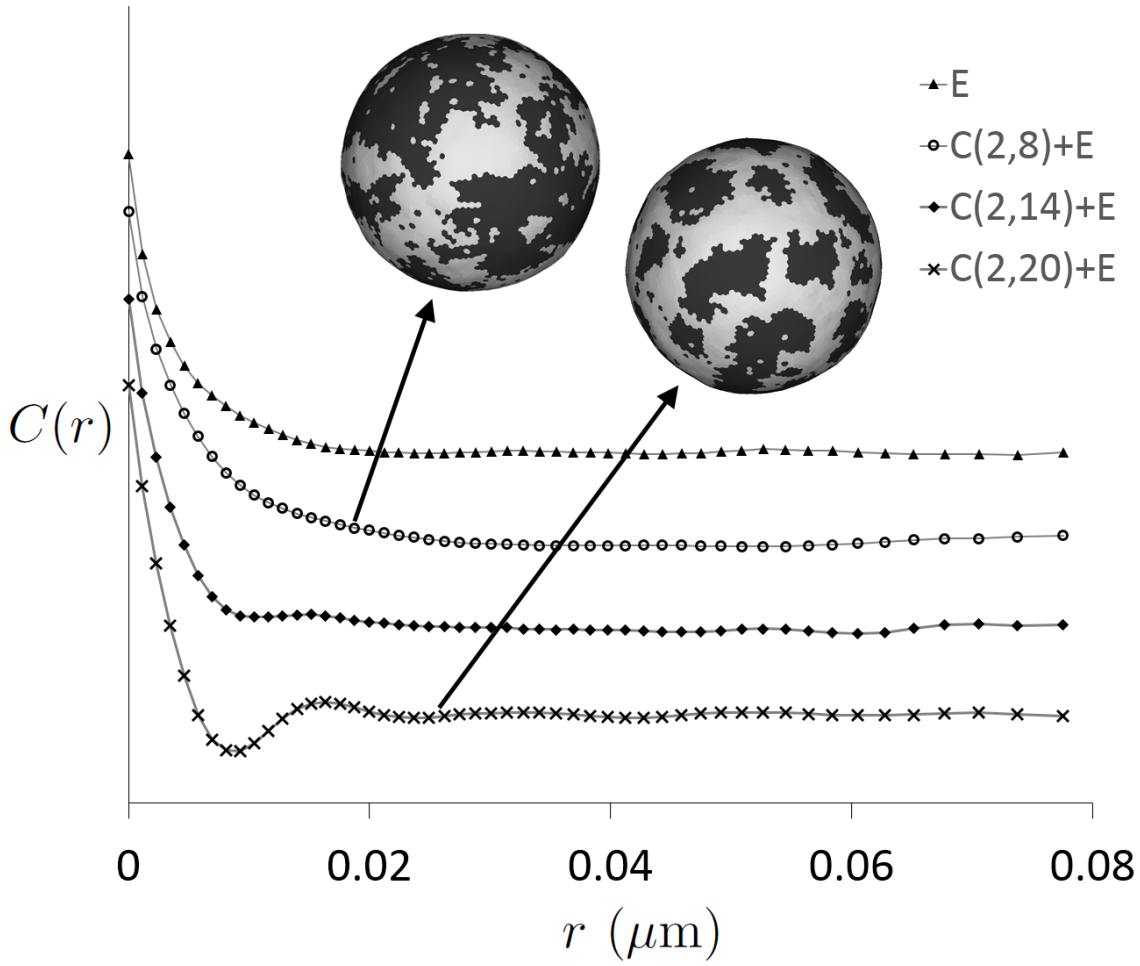


Figure 3.15: Curvature and electrostatics together stabilize modulated phases on LUVs. With modest values of bending moduli, curvature alone (not shown) yields macroscopic phase separation, while electrostatics alone (E) yields a roughly 9nm correlation length. Curvature with electrostatics (C(κ_d, κ_o)+E) yields a slight increase in correlation length, followed by the formation of modulated phases with domain diameter of 8nm as κ_o is increased. Correlation curves shown are stacked with a constant vertical offset.

Comparison to SANS Experiments

Recently the work of Heberle et. al [12] has shown that SANS experiments detect signatures of small domains on the surface of 60nm LUVs. These measurements revealed domains approximately 14nm in diameter in the model membrane system DSPC/POPC/chol.

With simulation parameters chosen to emulate these experimental conditions, we make direct comparison of the length scales of our simulated morphology with those measured by neutron scattering. For electrostatic decay length of $k^{-1} = 3$nm (no salt) and vesicle diameter of 60nm the simulated phase domains have a correlation length of $9 - 11$nm (inset of figure 3.14). This result is close to the direct size measurements of SANS, showing that electrostatic repulsion can compete with line tension to stabilize structures in the predicted size range of the nanodomains.

3.5 Conclusion

Expanding on the simulation techniques from our previous work [2], we have identified line tension as a running coupling constant, which necessitates the use of renormalization group techniques to properly account for how line tension must change as degrees of freedom are lost in coarse-graining. The renormalization procedure we outline can be used to relate the line tension at any two grain levels to produce equivalent morphologies (defined as having the same correlation length). Furthermore, this technique may be used as a general way of empirically determining renormalization flows for a given parameter.

Background curvature is found to play an important role in the formation of modulated phases. When parameters that give modulated phases on a sphere are used on a flat sheet (of equivalent area and vertex density) modulated phases do not appear. A series of simulations in which background curvature was varied show the transition to modulated phases only after a significant degree of background curvature is present.

We implement a model for the dipolar interactions between lipids and find that electrostatic repulsion can explain the formation of nanodomains in model membrane systems. Increasing the dipole density contrast, $\Delta\mu$, leads to break-up of macroscopic domains into irregular clusters. The nature of this transition is caused by a shift of the miscibility critical point to a higher value of line tension.

The decay length of the electric fields within the plane of the bilayer is found to be important to the phase morphology. Even though electrostatic interactions have a decay length of only a few nanometers, this is still several lipid shells, which proves important at the small scales relevant to nanodomains and membrane rafts. Varying the decay length from 1nm up to 3nm can drive a transition from macroscopic domains to clusters with correlation lengths on the order of 10nm.

Surprisingly, on a highly curved membrane (such as the surface of 60nm vesicles) the inclusion of both curvature and electrostatics can lead to a *larger* overall domain size than with electrostatic repulsion alone. This increase shows that curvature can work to either stabilize or to disperse domains, depending on the parameters and interactions used in the simulation. Further increasing the stiffness of the raft phase leads to the formation of modulated phases. This result is surprising because all bending moduli used in this study produce only macroscopic phase separation in the absence of the electrostatic interaction.

The size of simulated domains for 60nm LUVs can be compared to the experimentally measured domain sizes from SANS. The range of correlation lengths from simulations ($9 - 11$nm) matched well with the measured domain size in the model system DSPC/POPC/chol ($\approx$ 14nm diameter). Thus a dipolar repulsion can compete with line tension to stabilize phase domains on the nanometer scale.

Acknowledgements

Support was from research award NSF MCB 0842839 (to G.W.F.). J.J.A. received support from NIH Training Grant award 1-T32 GM 08267. We thank professor Benjamin Widom for help in formulating our ideas about renormalization and scaling.

Bibliography

[1] See Supplemental Material at [URL will be inserted by publisher] for proof of edge-case Guassian curvature equation and detailed explanation of decay length calculations.

[2] J. Amazon, S. Goh, and G. Feigenson. Competition between line tension and curvature stabilizes modulated phase patterns on the surface of giant unilamellar vesicles: a simulation study. *Phys. Rev. E.*, 87:022708, 2013.

[3] A. Ayuyan and F. Cohen. Raft composition at physiological temperature and ph in the absence of detergents. *Biophysical Journal*, 94:2654–2666, 2008.

[4] D. Brown. Lipid rafts, detergent-resistant membranes, and raft targeting signals. *Physiol.*, 21:430–439, 2006.

[5] A. Demchenko and S. Yesylevskyy. Nanoscopic description of biomembrane electrostatics: Results of molecular dynamics simulations and fluorescence probing. *Chem. and Phys. of Lipids*, 160:63–84, 2009.

[6] S. Duane and R. Horgan. Block spin techniques applied to the ising model. *Z. Phys. C*, 16:355–359, 1983.

[7] M. Ediden. Lipids on the frontier: a century of cell-membrane bilayers. *Nature Rev.*, 4: 414–418, 2003.

[8] G. Feigenson and J. Buboltz. Ternary phase diagram of dipalmitoyl-pc/dilauroyl-pc/cholesterol: nanoscopic domain formation driven by cholesterol. *Biophysical Journal*, 80:2775–2788, 2001.

[9] S. Goh, J. Amazon, and G. Feigenson. Toward a better raft model: Modulated phases in the four-component bilayer, dspc/dopc/popc/chol. *Biophysical Journal*, 104:853–862, 2013.

[10] Jay T. Groves. Bending mechanics and molecular organization in biological membranes. *Annu. Rev. Phys. Chem.*, 58:697–717, 2007.

[11] F. Heberle, J. Wu, S. Goh, R. Petruzielo, and G. Feigenson. Comparison of three ternary lipid bilayer mixtures: Fret and esr reveal nanodomains. *Biophysical Journal*, 99:3309–3318, 2010.

[12] F. Heberle, R. Petruzielo, J. Pan, P. Drazba, N. Kucerka, R. Standaert, G. Feigenson, and J. Katsaras. Bilayer thickness mismatch controls domain size in model membranes. *J. Am. Chem. Soc.*, 135:6853–6859, 2013.

[13] W. Helfrich and R. Servuss. Undulations, steric interaction and cohesion of fluid membranes. *Il Nuovo Cimento*, 3:137–151, 1984.

[14] A. Honerkamp-Smith, P. Cicuta, M. Collins, S. Veatch, M. Nijs, M. Schick, and S. Keller. Line tensions, correlation lengths, and critical exponents in lipid membranes near critical points. *Biophysical Journal*, 95:236–246, 2008.

[15] R. Houtappel. Order-disorder in hexagonal lattices. *Physica*, 16:425–255, 1950.

[16] J. Hu, T. Weikl, and R. Lipowsky. Vesicles with multiple membrane domains. *Soft Matter*, 7:6092, 2011.

[17] M. Hu, J. Briguglio, and M. Deserno. Determining the gaussian curvature modulus of lipid membranes in simulations. *Biophysical Journal*, 102:1403–1410, 2012.

[18] A. Imperio and L. Reatto. Microphase separation in two-dimensional systems with competing interactions. *J. Chem. Phys.*, 124:164712, 2006.

[19] T. Kawakatsu, D. Andelman, K. Kawasaki, and T. Taniguchi. Phase transitions and shapes of two component membranes and vesicles i: Strong segregation limit. *J. Phys. II*, 3:971–997, 1993.

[20] H. Kleinert. Thermal softening of curvature elasticity in membranes. *Phys. Lett.*, 114A: 263–268, 1986.

[21] T. Konyakhina, S. Goh, J. Amazon, F. Heberle, J. Wu, and G. Feigenson. Control of nanoscopic-to-macroscopic transition: modulated phases in four-component dspc/dopc/popc/chol giant unilamellar vesicles. *Biophysical Journal*, 101:L08–L10, 2011.

[22] I. Levental, M. Grzybek, and K. Simons. Raft domains of variable properties and compositions in plasma membrane vesicles. *PNAS*, 108:11411–11416, 2011.

[23] D. Lingwood and K. Simons. Lipid rafts as a membrane-organizing principle. *Science*, 327:46–50, 2010.

[24] J. Liu, S. Qi, J. Groves, and A. Chakraborty. Phase segregation on different length scales in a model cell membrane system. *J. Phys. Chem. B*, 109:19960–19969, 2005.

[25] J. Liu, S. Qi, J. Groves, and A. Chakraborty. Phase segregation on different length scales in a model cell membrane system. *J. Phys. Chem. B*, 109:19960–19969, 2005.

[26] B. Machta, S. Papanikolau, J. Sethna, and S. Veatch. Minimal model of plasma membrane hterogeneity requires coupling cortical actin to criticality. *Biophysical Journal*, 100:1668–1677, 2011.

[27] D. Marsh. Elastic curvature constants of lipid monolayers and bilayers. *Chem. and Phys. of Lipids*, 144:146–159, 2006.

[28] S. Munro. Lipid rafts: elusive or illusive? *Cell*, 115:377–388, 2003.

[29] H. Nymeyer and H. Zhou. A method to determine dielectric constants in nonhomogeneous systems: Application to biological membranes. *Biophysical Journal*, 94:1185–1193, 2008.

[30] I. Parmryd and B. Onfelt. Consequences of membrane topography. *FEBS J.*, 280: 2775–2784, 2013.

[31] U. Peterson, D. Mannock, R. Lewis, P. Pohl, R. McElhaney, and E. Pohl. Origin of membrane dipole potential: Contribution of the phospholipid fatty acid chains. *Chem. and Phys. of Lipids*, 117:19–27, 2002.

[32] L. Pike. Lipid rafts: bringing order to chaos. *J. of Lipid Research*, 44:655–667, 2003.

[33] A. Raudino and D. Mauzerall. Dielectric properties of the polar head group region of zwitterionic lipid bilayers. *Biophysical Journal*, 50:441–449, 1986.

[34] S. Semrau, T. Idema, L. Holtzer, T. Schmidt, and C. Storm. Accurate determination of elastic parameters for multicomponent membranes. *Phys. Rev. Lett.*, 100:088101, 2008.

[35] P. Sengupta, D. Holowka, and B. Baird. Flourescence resonance energy transfer between lipid probes detects nanoscopic heterogeneity in the plasma membrane of live cells. *Biophysical Journal*, 92:3564–3574, 2007.

[36] M. Seul and D. Andelman. Domain shape and patterns: the phenomenology of modulated phases. *Science*, 267:476–483, 1995.

[37] K. Simons and G. van Meer. Lipid sorting in epithelial cells. *Biochemistry*, 27:6197–6202, 1988.

[38] K. Simons and W. Vaz. Model systems, lipid rafts, and cell membranes. *Annu. Rev. Biophys. Biomol. Struct.*, 33:269–295, 2004.

[39] A. Smondyrev and M. Berkowitz. Molecular dynamics simulation of dipalmitoylphosphatidylcholine membrane with cholesterol sulfate. *Biophysical Journal*, 78:1672–1680, 2000.

[40] M. Swamy, L. Ciani, M. Ge, A. Smith, D. Holowka, B. Baird, and J. Freed. Coexisting domains in the plasma membranes of live cells characterized by spin-label esr spectroscopy. *Biophysical Journal*, 90:4452–4465, 2006.

[41] A. Travesset. Effect of dipolar moments in domain size of lipid bilayers and monolayers. *J. Chem. Phys.*, 125:084905, 2006.

[42] T. Ursell, W. Klug, and R. Phillips. Morphology and interaction between lipid domains. *PNAS*, 106:13301–13306, 2009.

[43] S. Veatch and S. Keller. Seeing spots: complex behavior in simple membranes. *Biochim. Biophys. Acta.*, 1746:172–185, 2005.

[44] L. Wang. Measurements and implications of the membrane dipole potential. *Annu. Rev. Biochem.*, 81:615–635, 2012.

[45] L. Wang, P. Bose, and F. Sigworth. Using cryo-em to measure the dipole potential of a lipid membrane. *PNAS*, 103:18528–18533, 2006.

[46] J. Zhao, J. Wu, F. Heberle, T. Mills, P. Klawitter, G. Huang, G. Costanza, and G. Feigenson. Phase studies of model biomembranes: complex behavior of dspc/dopc/cholesterol. *Biochim. Biophys. Acta.*, 1768:2764–2776, 2007.

Chapter 4

Conclusion and Future Studies

4.1 Conclusion

4.1.1 Lattice Simulations of Modulated Phases

In the four-component model mixture DSPC/DOPC/POPC/chol modulated phases were observed in a small region of the composition space (figure 2.1). These patterns are static on the GUV surface and stable over long time scales. To model the formation of this novel phase morphology a competing interactions model was used (section 2.3.2). The competing interaction model states that when a single order parameter (in this case a phase field 2.3.3) is subject to multiple interactions a stable modulated phase can be formed. For this study line tension and curvature fields (sections 2.3.4 and 2.3.5) were used as the competing interactions. The energetics were modeled using the Helfrich energy functional with appropriate constraints as defined in section 2.3.6.

This model was implemented in a lattice framework, in which the GUV was approximated as a discrete triangular lattice. The line tension term was computed by interpolating the arc length of segments between adjacent vertices (section 2.4.2). The discrete mean and

Gaussian curvatures were computed using the algorithms outlined in sections 2.4.3 and 2.4.4 respectively. The discrete Hamiltonian that includes the energetic contributions of line tension and curvature was minimized by a Monte Carlo simulation (section 2.4.6) that uses the Metropolis-Hastings algorithm to determine the equilibrium morphology.

For a certain set of parameters (table 2.1) this simulation was able to reproduce the existence of modulated phases observed on the surface of GUVs (section 2.5.1). The resulting simulation morphologies closely matched many of the observed modulated phase patterns (figure 2.7). These patterns were shown to be more thermodynamically stable (section 2.5.2) than other common morphologies observed on GUVs, such as a single macroscopic domains. The parameters were found to dictate the details of the modulated phase patterns (section 2.5.3). The phase fraction determined whether there were honeycomb or stripe like patterns (figure 2.10), the line tension changed the thickness of the pattern and ultimately produced macroscopic domains (figure 2.11), and the bending modulus decreased the repeat length of the patterns (figure 2.12) as the bending modulus of L_o was increased. The role of the subtle background curvature present on a spherical lattice was also explored (section 3.3.2). A flat lattice was constructed and (using the same parameters that produced modulated phases on a spherical lattice) it was shown that modulated phases do not appear on the flat lattice (section 3.4.2).

4.1.2 Lattice Simulations Electrostatics on LUVs

To better understand the nature of nanodomains in a model mixture such as DSPC/POPC/chol a competing interactions model is again used. Instead of curvature (which has proven useful in explaining micron size morphology features), the electrostatic dipole repulsion between adjacent lipid molecules was considered to compete with line tension. The electrostatic environment of the membrane was modeled by treating each vertex as if it had a dipole density

pointing normal to the membrane (section 3.3.4). The electric field was then computed at each point by summing over the proper kernel (equation 3.5). An exponential decay factor is included to emulate the effects of electrostatic screening from ions in solution. The electrostatic decay length due to screening was computed using the methods outlined in appendix B and was found to be in the range $(1-4)$nm under all salt conditions.

Using this model on LUVs and the parameters outlined in table 3.2 it was shown that electrostatic repulsion under physiological conditions was sufficient to compete with line tension and break apart macroscopic domains (section 3.4.3). The contrast between the dipole density of the two phases was shown to cause an abrupt transition from macroscopic to smaller clusters with diminishing correlation length (figure 3.12). This abrupt transition was found to be caused by a shift of the critical point, as observed in the heat capacity curve, to higher line tension (figure 3.13). The electrostatic decay length, which is dictated by the concentration of ions in the surrounding medium, was also found to cause break up of macroscopic domains (figure 3.14) over its range of accessible values.

One unexpected result was that when electrostatics and curvature are included together, using experimentally measured values of bending modulus ($\kappa_d = 2\times10^{-19}$J, $\kappa_o = 8\times10^{-19}$J), the correlation length dilates slightly from 9nm to 12nm. However, if the stiffness of L_o is further increased we find the formation of modulated phases with domains 8nm in diameter. These results are summarized in figure 3.15. The significance of this finding is that: **(1)** Curvature does not solely oppose line tension, as indicated by the dilation of correlation length; **(2)** In the absence of electrostatics, the values of bending modulus used would show macroscopic domains; likewise, in the absence of curvature the morphologies would be irregular clusters with a correlation length of 9nm; only when both interactions are used in tandem do we find modulated phases at unexpected parameter values.

The correlation lengths measured in the electrostatics simulations were compared to the measured domain sizes as determined by Heberle et al. using SANS. It was found that the

size scales observed in simulations of around $(9-11)$nm matched the domain sizes reported by SANS of approximately 14nm in diameter. This shows that electrostatics may play a role in the stabilization of nanodomains in model systems and on the plasma membrane.

4.1.3 Renormalization of Line Tension

The values of line tension used in the simulation of modulated phases on GUVs in chapter 2 were smaller than any experimentally reported values. The reason for this was found to be the effect of coarse graining on line tension. This was solved using renormalization to relate the values of line tension at arbitrary grain levels (section 3.4.1). A scale invariant form of line tension u was defined, which assumed that there is a true physical value of line tension that is only valid at the microscopic scale. The scale invariant line tension scales as $\sqrt{G}$ which explains why the raw line tension must be so small on highly coarse grained lattices to preserve the value of u. The scale invariant line tension also proved useful in relating this model to the triangular lattice Ising model (figure 3.6) and rigorously defining the critical point.

Preserving the value of u did not produce equivalent morphologies across grain levels as shown in figure 3.9. To solve this problem a size measure, the inverse correlation length $c(u/kT)$ (equation 3.13), for morphological features was defined and two morphologies were defined to be equivalent if they had the same value of inverse correlation length. The condition that two morphologies at different grain levels have the same inverse correlation length defines a scaling relation (equation 3.14). This equation was numerically solved to give rise to a set of renormalization flows for the values of u (figure 3.8). This calculation shows that under coarse graining, u increases if we start above the critical line tension and decreases if we start below. The value of u remains fixed at the critical point, consistent with self-similar fluctuations on all length scales.

4.2 Future Directions

4.2.1 Topography of Living Cells

The simulation outlined in this body of work has to this point been used to study phase morphology on the surface of spheres and flat sheets of membrane. While these topologies are useful in considering simplified systems, such as GUVs or sections of model membranes, they do not represent the full repertoire of topography present on a living cell. To capture the effects of more exotic membrane shapes we plan to construct lattices that represent highly curved structures common on the surface of living cells. The work of Parmryd et al. has shown that the surface of a living cell is not smooth as most models depict [8], but instead covered in pits, folds, and bumps. These topographical features suggest that the plasma membrane has approximately four times as much area as would be expected by its enclosed volume in some cell types.

Given the observation that curvature plays a significant role in dictating morphology we will use the lattice simulation to study phase sorting on various model structures known to exist on the surface of the membrane. The topographical features that we will be looking at (but not limited to) are **1)** Spherical vesicles **2)** Flat planar membranes **3)** Spherical caps of varying curvature **4)** Tubular membrane sections **5)** Microvilli structures in various stages of development **6)** Tight clusters of microvilli and **7)** Budding vesicles. In addition to these idealized features we will also import height maps of real cells measured using scanning conductance microscopy [7].

Using these topographies we will perform simulations on each structure for a wide range of parameters and catalogue the phase morphology. This will allow us to identify regions of interest within the parameter space and form a library of images that cell biologists will be able to use to get an intuition for how curvature on the cell and lipid rafts may be related. The suggested parameter ranges that we will be exploring are given in table 4.1.

parameter	value	unit
κ_d	200	z
κ_o	$[200, 20000]$	z
$\bar{\kappa}_d$	-200	z
$\bar{\kappa}_o$	$[-20000, -200]$	z
C_d	0	μm^{-1}
C_o	$[-10, 10]$	μm^{-1}
μ_d	133	$e^-/\mu\text{m}$
μ_o	309	$e^-/\mu\text{m}$
k^{-1}	0.002	μm
u/kT	$[0, 0.75]$	N/A
R	$[0.03, 50]$	μm
P	$[0, 1]$	N/A
kT	4.0	z

Table 4.1: Proposed parameter ranges for exploration of phase morphology on topographical features of cells. ($z = 10^{-21}$ J)

In addition to the library of phase morphologies that we plan to compile, this study would seek to address a contradiction between our simulation results and experimental observations on living cells. When microvilli are studied it is found that they contain primarily raft lipids (sphingolipids and cholesterol) [6, 11, 2], whereas in our simulation we would find L_d on such highly curved structures to minimize the energy cost of bending the membrane. Can we induce phase sorting of L_o to highly curved regions of membrane using some set of parameters in our present model? One possible explanation may be the leaflet asymmetry of the plasma membrane which, unlike symmetric model systems, would produce a non-zero spontaneous curvature (C_d,C_o). This possibility will be explored.

4.2.2 Protein Fields

To this point this model has been used to explain phase morphology in lipid only systems while neglecting the possible interactions of membrane proteins. Current estimates of the protein content of the plasma membrane place it at 15% of the bilayer volume, so a better

model of the plasma membrane should include a significant protein component. The entire proteome of the plasma membrane is too complicated to model at a species level, however we may ask what effects a generalized protein component [9, 10, 12] of the membrane may have on phase morphology (and by extension lipid rafts).

To model the effects of a protein component on phase morphology we include a protein concentration field, p_v, in our simulation model. This is a field that takes on continuous values defined on the vertices of the lattice with units of area concentration (mol/μm^2). To model its interaction with the bulk of the membrane we use the simplest possible bilinear coupling.

$$\mathcal{H}_{\text{prot}} = \sum_v a(\phi_v) p_v \Delta A_v \tag{4.1}$$

$$a(\phi) = \begin{cases} a_d, & \phi \in L_d \\ a_o, & \phi \in L_o \end{cases} \tag{4.2}$$

The coupling, $a(\phi_v)$, is a piece wise function that depends on the local phase of vertex v. This term has units of z/mol and sets the free energy difference for having a given amount of protein in each phase. The difference in this coupling is related to the partition coefficient of the protein, K_p, which can be measured experimentally,

$$a_o - a_d = N_A kT \ln(K_p) \tag{4.3}$$

where N_A is Avogadro's constant. To keep the protein concentration from dipping below zero or approaching infinity a non-linear constraint (threshold linear function) is imposed to ensure that $p_v \in [0, p_{\max}]$, where $p_{\max}$ is the maximum protein solubility in the membrane. The total amount of protein is also assumed fixed, but this constraint is imposed by perform-

ing exchanges of protein concentration between vertices (similar to how the phase fraction is held fixed).

Another key measurement that will be of use when proteins are added to the membrane will be the shift in the rho window. A shift in the modulated phase window to higher ρ indicates that more DOPC must be added to achieve the same line tension and thus line tension must have decreased, whereas a shift to lower ρ implies that line tension must have increased. This identification of line tension with a shift in the ρ-window may be able to indicate if the protein partitions to or away from the interface.

To model the proteins effect on the interface, a virtual line tension must be included that depends on the concentration of the protein near the interface.

$$\mathcal{H}_{\text{prot-line}} = \sum_{v} \gamma_{\text{prot}} p_v L_v \tag{4.4}$$

The new line tension γ_{prot} is a protein field dependent line tension with units of ($z\mu$m/mol) and L_v is the contribution to phase boundary of vertex v. The sign of γ_{prot} dictates whether or not proteins partition to ($\gamma_{\text{prot}} < 0$) or away from ($\gamma_{\text{prot}} > 0$) the interface. This model has some limitations due to the coarse-grained nature of the lattice simulation. We cannot resolve below the inter-vertex spacing (on the order of 1μm for GUV simulations) so partitioning of the protein around the interface is limited to that scale.

4.2.3 Asymmetric Membranes

The plasma membrane is known to have distinct lipid compositions on its inner and outer leaflet [13]. The outer leaflet is composed primarily of phosphatidylcholine, sphingomyelin, and cholesterol, while the inner leaflet is composed of phosphatidylethanolamine, phosphatidylserine, phosphatidylinositol, and cholesterol [3]. While models mimicking the outer leaflet have proven useful in studying lipid phase behavior [5], inner leaflet models have never

yet produced detectable phase coexistence [14]. This is a problem, because many models postulate that virus assembly and budding occur at raft regions of the inner leaflet [4]. Since rafts are well evidenced on the outer leaflet one hypothesis is that coupling between the leaflets can precipitate rafts in the inner leaflet [1].

To study the coupling of the two leaflets we introduce a second phase field to the simulation model. The simplest Hamiltonian that captures this behavior is a piece wise function that specifies interaction energies for each possible combination of phases in the two leaflets.

$$\mathcal{H}_{\text{leaflet}} = \sum_{v} b(\phi_v^{(\text{outer})}, \phi_v^{(\text{inner})}) \Delta A_v \tag{4.5}$$

$$b(\phi^{(\text{outer})}, \phi^{(\text{inner})}) = \begin{cases} b_{o-o}, & \phi^{(\text{outer})} \in L_o, \phi^{(\text{inner})} \in L_o \\ b_{o-d}, & \phi^{(\text{outer})} \in L_o, \phi^{(\text{inner})} \in L_d \\ b_{d-o}, & \phi^{(\text{outer})} \in L_d, \phi^{(\text{inner})} \in L_o \\ b_{d-d}, & \phi^{(\text{outer})} \in L_d, \phi^{(\text{inner})} \in L_d \end{cases} \tag{4.6}$$

The phases in the outer and inner leaflet are given by the phase fields $\phi^{(\text{outer})}$ and $\phi^{(\text{inner})}$ respectively. The coupling function $b(\phi^{(\text{outer})}, \phi^{(\text{inner})})$ distinguishes between the four possible ways of mixing the inner and outer leaflet phases and has units of free energy per unit area ($z/\mu\text{m}^2$). Notice that b is not assumed to be a symmetric function. This expresses the fact that the phases formed by the outer and inner leaflet may be vastly different in their material properties.

The values of each b_{x-y} that define the coupling function may be measured experimentally to within a constant energy offset. To do this asymmetric GUVs will be prepared that have different colored dyes in each leaflet that partition into one of the two phases. The two channels can be measured independently and correlated to show where L_o and L_d phases

overlap on the two leaflets. By measuring the probability of each phase combination we can estimate the relative free energy per unit area and hence the coupling constants.

Bibliography

[1] D. Allender and M. Schick. Phase separation in bilayer lipid membranes: Effects on the inner leaf due to coupling to the outer leaf. *Biophysical Journal*, 91:2928–2935, 2006.

[2] R. Anderson and K. Jacobson. A role for lipid shells in targeting proteins to caveole, rafts, and other lipid domains. *Science*, 296:1821–1825, 2002.

[3] H. Cheng, Megha, and E. London. Preparation and properties of asymmetric vesicles that mimic cell membranes. *J. Biol. Chem.*, 284:6079–6092, 2009.

[4] M. Edidin. Lipids on the frontier: A century of cell-membrane bilayers. *Nature Rev. Mol. Cell Biol.*, 5:414–418, 2003.

[5] S. Goh, J. Amazon, and G. Feigenson. Toward a better raft model: Modulated phases in the four-component bilayer, dspc/dopc/popc/chol. *Biophysical Journal*, 104:853–862, 2013.

[6] E. Ikonen and K. Simons. Protein and lipid sorting from the trans-golgi network to the plasma membrane in polarized cells. *Sem. Cell Devel. Biol.*, 9:503–509, 1998.

[7] D. Klenerman, Y. Korchev, and S. Davis. Imaging and characterization of the surface of live cells. *Curr. Opin. Chem. Biol.*, 15:696–703, 2011.

[8] I. Parmryd and Bjorn Onfelt. Consequences of membrane topography. *febs.*, 280:2775–2784, 2013.

[9] M. Partenskii and P. Jordan. Membrane deformation and the elastic energy of insertion: Perturbation of membrane elastic constants due to insertion. *J. Chem. Phys.*, 117: 10768–10776, 2002.

[10] B. Reynwar, G. Illya, V. Harmandaris, M. Muller, K. Kremer, and M. DeSerno. Aggregation and vesiculation of membrane proteins by curvature-mediated interactions. *Nature*, 447:461–464, 2007.

[11] K. Simons and E. Ikonen. Functional rafts in cell membranes. *Nature*, 387:569–572, 1997.

[12] M. Sperotto, S. May, and A. Baumgartner. Modelling of proteins in membranes. *Chem. Phys. Lipids*, 141:2–29, 2006.

[13] J. Virtanen, K. Cheng, and P. Somerharju. Phospholipid composition of the mammalian red cell membrane can be rationalized by a superlattice model. *PNAS*, 95:4964–4969, 1998.

[14] T. Wang and J. Silvius. Cholesterol does not induce segregation of liquid-ordered domains in bilayers modeling the inner leaflet of the plasma membrane. *Biophysical Journal*, 81:2762–2773, 2001.

Appendix A

Supplemental Material

A.1 Chapter 2 Supplement

A.1.1 Proof of scaling relations

Let $S = \sigma(u, v)$ be a parameterized surface patch in three dimensions. The first and second fundamental form of this surface are defined as,

$$E = \vec{\sigma}_u \cdot \vec{\sigma}_u \quad F = \vec{\sigma}_u \cdot \vec{\sigma}_v \quad G = \vec{\sigma}_v \cdot \vec{\sigma}_v \tag{A.1}$$

$$L = \vec{\sigma}_{uu} \cdot \vec{n} \quad M = \vec{\sigma}_{vu} \cdot \vec{n} \quad N = \vec{\sigma}_{vv} \cdot \vec{n}, \tag{A.2}$$

where the subscripts denote partial derivatives and $\vec{n}$ is the surface normal at that point. The mean and Gaussian curvature at every point can then be defined in terms of the first and second fundamental forms as,

$$H = \frac{LG - 2MF + NE}{2(EG - F^2)} \tag{A.3}$$

$$G = \frac{LN - M^2}{EG - F^2} \tag{A.4}$$

Now consider a linear scaling of the surface, $S' = aS$. This has the following scaling effects on the first and second fundamental forms,

$$E' = a\vec{\sigma}_u \cdot a\vec{\sigma}_u = a^2 E \quad F' = a\vec{\sigma}_u \cdot a\vec{\sigma}_v = a^2 F \quad G' = a\vec{\sigma}_v \cdot a\vec{\sigma}_v = a^2 G \tag{A.5}$$

$$L' = a\vec{\sigma}_{uu} \cdot \vec{n} = aL \quad M' = a\vec{\sigma}_{vu} \cdot \vec{n} = aM \quad N' = a\vec{\sigma}_{vv} \cdot \vec{n} = aN \tag{A.6}$$

The normal vector, $\vec{n}$, remains unchanged under a linear scaling. Thus the mean and Gaussian curvature of S' are given by,

$$H' = \frac{L'G' - 2M'F' + N'E'}{2(E'G' - F'^2)} = \frac{a^3(LG - 2MF + NE)}{2a^4(EG - F^2)} = \frac{H}{a} \tag{A.7}$$

$$G' = \frac{L'N' - M'^2}{E'G' - F'^2} = \frac{a^2(LN - M^2)}{a^4(EG - F^2)} = \frac{G}{a^2} \tag{A.8}$$

The area element of the surface scales similarly,

$$dA = \sqrt{EG - F^2}dudv \tag{A.9}$$

$$dA' = \sqrt{E'G' - F'^2}dudv = \sqrt{a^4(EG - F^2)}dudv = a^2 dA \tag{A.10}$$

Lastly, let the boundary between the L_d/L_o phases be given by the parametric curve $\vec{\gamma}(t) = \vec{\sigma}(u(t), v(t))$. The arc length of this curve scales as,

$$l = \oint_0^1 \sqrt{E\dot{u}^2 + 2F\dot{u}\dot{v} + G\dot{v}^2}dt \tag{A.11}$$

$$l' = \oint_0^1 \sqrt{E'\dot{u}^2 + 2F'\dot{u}\dot{v} + G'\dot{v}^2}dt = \oint_0^1 a\sqrt{E\dot{u}^2 + 2F\dot{u}\dot{v} + G\dot{v}^2}dt = al \tag{A.12}$$

Since we have made no assumptions about the geometry of the surface S, this general scaling relation is true for any surface geometry for which the first and second fundamental forms can be calculated.

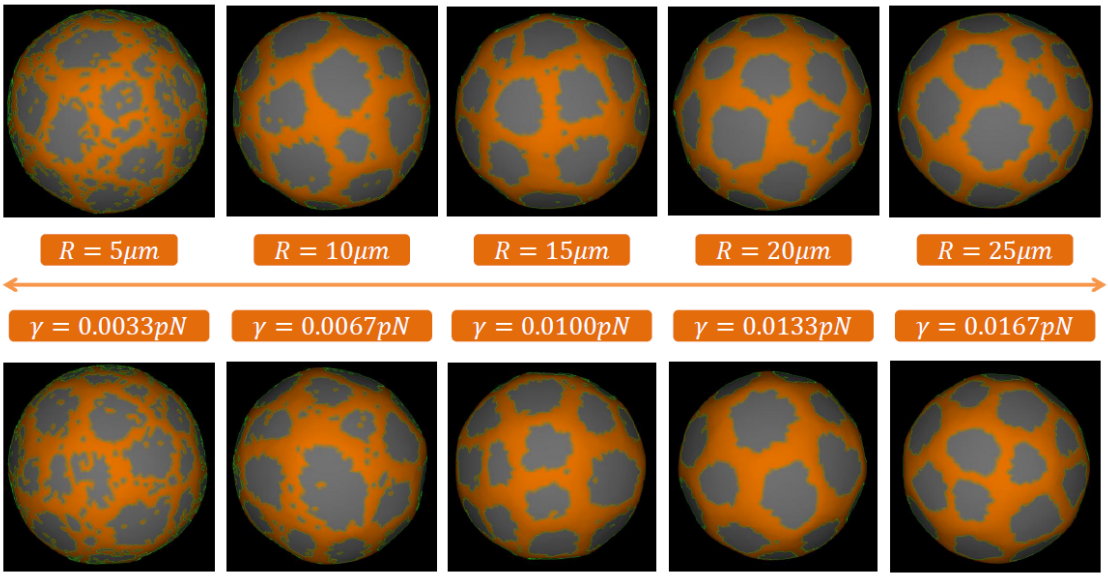


Figure A.1: Comparison of morphology over a trace of line tension and vesicle size. The relevant scaling variable is γR, which defines an equivalence class of phase morphology. Top trace holds line tension fixed at $\gamma = 0.01$pN; bottom trace holds radius fixed at $R = 15\mu$m.

We have performed simulations to test this scaling relation. Shown below are two series of simulations, one varying the size of the GUV (scaling the radius) and one varying the line tension.

In the top row of figure A.1 the line tension is fixed at $\gamma = 0.01$pN. In the bottom row the radius is fixed at 15μm. The series shows that scaling the size of the GUV has the same effect on the morphology as scaling the line tension by that same magnitude. All other parameters are as specified in Table I in the text.

A.1.2 Special case: small domain on a large vesicle

One thought experiment that would seem to contradict our scaling relation is that a small highly curved domain that is embedded in an effectively flat (very large vesicle) membrane would experience very little perturbation upon increasing the radius of the vesicle. This is

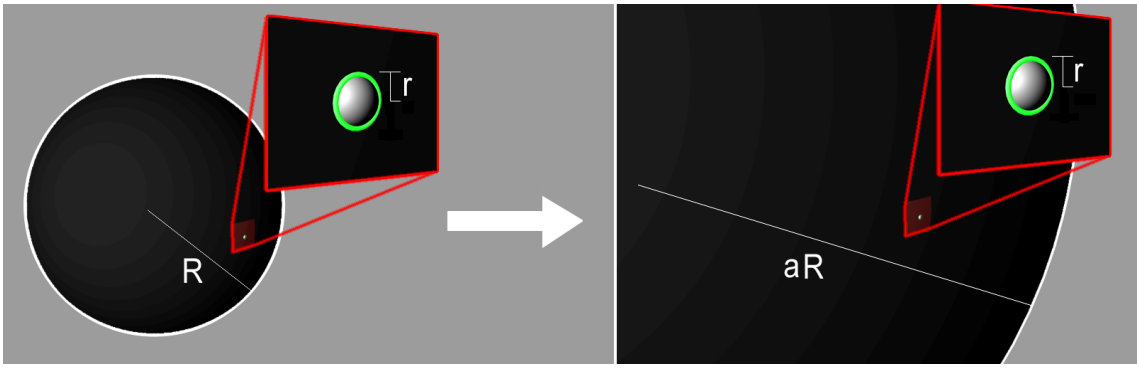


Figure A.2: Scaling scenario in which the radius of a vesicle is dilated by a factor a, but the size of a small phase domain is held fixed. This process does not preserve area fraction.

only true however if the scaling of the vesicle has no effect on the size of the domain or the amount of phase present. We explore this in detail here.

Consider a vesicle of radius R with a small highly curved L_d domain on its surface that has size $r << R$, and suppose that this is the minimal energy configuration. This domain would experience a relatively flat environment and not be highly influenced by the bulk curvature of the sphere. To express the energetics of this configuration, we break the integral up into two parts, one for the bulk L_o region of the vesicle and one for a small patch of area, $\Delta A << R^2$ centered on the small domain (the shaded red region shown below).

The energy of this vesicle may then be expressed as,

$$E = 8\pi\kappa_o - \frac{2\Delta A\kappa_o}{R^2} + \gamma L + C \approx 8\pi\kappa_o + \gamma L + C \tag{A.13}$$

Where L is the domain perimeter (shown in green) and C is the curvature terms integrated over the red surface patch. Since $\Delta A << R^2$ we may drop the second term with minimal perturbation to the energy (this expresses the fact that the surface is roughly planar in the vicinity of the domain).

Now suppose we scaled the vesicle, but did not change the dimensions/shape of the domain (except to ensure that it conformed to the subtle change in background curvature). This is shown in the second panel above. Notice that the red region was not scaled; we are assuming that $\Delta A << a^2R^2$ to ensure that the surface is still locally planar near the domain. The energy of this new vesicle is,

$$E' \approx 8\pi\kappa_o + \gamma L' + C' \tag{A.14}$$

The change in domain perimeter and the curvature energies in the small red patch would be minimal, as this region remained largely unchanged under the transformation, $L' \approx L$ and $C' \approx C$. Thus we find that,

$$E' \approx 8\pi\kappa_o + \gamma L' + C' \approx 8\pi\kappa_o + \gamma L + C \approx E \tag{A.15}$$

Since the original vesicle was assumed at equilibrium and the scaled vesicle has exactly the same energy landscape we would expect this to be the minimal energy configuration of the scaled vesicle. Therefore a small highly curved domain with a roughly planar background will not change under scaling, given that the domain itself is kept the same size/shape.

The phase fraction under the scaling outlined above is not held fixed, $P' = P/a^2$. In the scaling relation we put forth in chapter 2, we implicitly assume that the phase fraction is held fixed upon scaling, as shown below.

Under this scaling the area fraction of the phases remains constant and the energy scales to,

$$E' \approx 8\pi\kappa_o + a\gamma L + C \tag{A.16}$$

The extra factor of 'a' in the line tension portion shifts the energy landscape in some non-

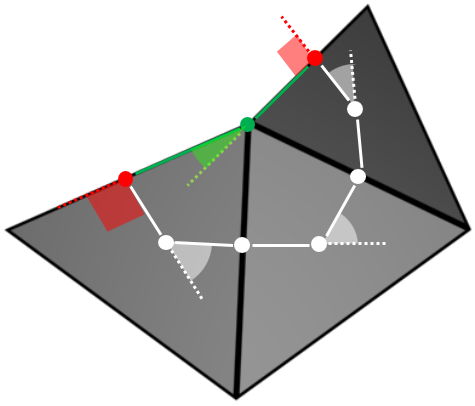
Figure A.3: Scaling scenario in which the radius of a vesicle is dilated by a factor a, and the size of a small phase domain is also dilated by a factor a. This process does preserve area fraction.

trivial way and would imply that the scaled domain is not necessarily the minimal energy configuration and may take on some other morphology if we re-equilibrate the scaled vesicle. The argument outlined in the paper is that the new equilibrium morphology is exactly the same as if we had kept the same size vesicle and increased the line tension by the factor 'a'. Stated another way, for any real positive constant, z, the relation $\gamma R = z$ defines an equivalence class of equilibrium morphologies across all vesicle sizes (assuming all other parameters are held fixed).

A.2 Chapter 3 Supplement

A.2.1 Proof of Gaussian curvature approximation

The equation presented in the manuscript (equation 3) satisfies the Gauss-Bonnet theorem. We show this by constructing the Voronoi region of a vertex at the edge of a triangulated surface and then using the Gauss-Bonnet theorem to derive equation 3. Consider the following geometry,



This is a generalized case for the surrounding geometry of a vertex (green circle) at the edge of the surface. All of the relevant angle deflections are highlighted. The Gauss-Bonnet theorem states that around this closed loop the following must hold true,

$$AG_{\text{ave}} + \sum \Delta\theta = 2\pi$$

The first term is the average Gaussian curvature (to be used as an approximation of G at the green vertex) within the closed loop times the area (A) of the Voronoi region. The second term is the sum of angle deflections where each segment of the defining contours meet (we assume that the Voronoi region is defined using arcs of geodesics so that the geodesic curvature is zero along the boundary). This equation is set equal to 2π because the region of interest is topologically equivalent to a plane, which has Euler characteristic of $\chi = 1$.

The terms in the angle deflection sum are highlighted in the figure above. The white angles can be shown to be equal to the interior angle of the triangle at the green vertex and are summed as for the normal full-loop Gaussian curvature computation. The interesting terms occur at the edges (red circles) and at the green vertex where the two edges that define the open edge of the surface intersect. The two red circles contribute $\pi/2$ each because the

arcs always meet an edge at a right angle. The angle deflection that occurs at the green point needs to be treated carefully. The angle between two intersecting arcs on a curved surface is always defined by projecting the arcs into the tangent plane of the surface at the point of intersection and measuring the angle in this plane. To perform this we use the normal at the apex point (defined as the area-weighted sum of the adjacent faces normals) to project the two edges shown in green into the tangent plane. This is precisely the construction shown in figure 3 of the manuscript. Once this is done we measure the angle deflection at the apex (light green angle). The angle deflection sum is given by,

$$\sum \Delta\theta = \sum \theta_{\text{white}} + \pi + \theta_{\text{green}}$$

With some quick geometry it can be shown that $\theta_{\text{green}} = \pi - \theta_P$. This gives a resultant equation for the Gaussian curvature of,

$$G_{\text{ave}} = \frac{2\pi - \theta_P - \sum \theta_{\text{white}}}{A}$$

This is the expression given in the manuscript.

A.2.2 Poisson-Boltzmann Equation Simulation

See appendix B.

A.2.3 Error Estimates for Discrete Approximations

Here we present test cases for error estimation of our discrete approximations.

Sphere (R = 10.0)

quantity	theoretical value	simulation value	error
$\langle H \rangle$	0.1	0.101126	1.1%
$\langle G \rangle$	0.01	0.010119	1.2%
$\iint H^2 dA$	4π	12.680144	0.91%
$\iint G dA$	4π	12.573484	0.06%

Sphere after 10,000 MC sweeps. (R = 10.0)

quantity	theoretical value	simulation value	error
$\langle H \rangle$	0.1	0.101077	1.07%
$\langle G \rangle$	0.01	0.011991	1.2%
$\iint H^2 dA$	N/A	101.002129	N/A
$\iint G dA$	4π	12.573351	0.06%

Flat Sheet (R = 10.0)

quantity	theoretical value	simulation value	error
$\langle H \rangle$	0.0	0.000925	N/A
$\langle G \rangle$	0.0	0.00051	N/A
$\iint H^2 dA$	0	0.000675	N/A
$\iint G dA$	0	0.092991	N/A

To test arc-length estimation, a circular domain that subtends the given angle is drawn on a sphere and its circumference measured.

Circular domain boundary (R = 25.0)

domain angle	theoretical value	simulation value	ratio
π	157	203	0.77
0.87π	153	196	0.78
0.74π	143	183	0.79
0.59π	125	196	0.78
0.41π	94.2	122	0.77

The actual error in arc length is quite large ($\approx 30\%$). This is expected because we are approximating smooth arcs on a discrete surface. The ratio of the measured and expected values for these distances tells us how much distortion is due to the discrete lattice. If our approximation is consistent, then this proportion should be roughly constant. We find this to be true with a distortion constant of approximately 0.78.

Appendix B

Poisson-Boltzmann Equation

To better understand the electrostatic environment of the lipid bilayer we turn to the Poisson-Boltzmann equation, which expresses the electrostatic potential in terms of the charge distribution, spatially varying dielectric constant [3], and a spatially varying screening length. In practice we use a linearized version of the Poisson-Boltzmann equation known as the Debye-Huckel equation which can be written in its entirety as,

$$\nabla \cdot [\epsilon(\vec{r})\nabla V(\vec{r})] = -\rho(\vec{r}) + \epsilon(\vec{r})\kappa^2(\vec{r})V(\vec{r}) \tag{B.1}$$

Where $\epsilon(\vec{r})$ is the dielectric constant, $V(\vec{r})$ is the electrostatic potential, $\rho(\vec{r})$ is the charge density distribution, and $\kappa(\vec{r})$ is one over the electrostatic screening length (Debye length). This screening length can be defined in terms of the concentrations of different ion species, C_i, in the surrounding medium by the equation,

$$\kappa^2 = \frac{e^2}{\epsilon kT}\sum_i z_i^2 C_i \tag{B.2}$$

Where z_i is the charge number of ion species i.

To numerically solve for the electrostatic potential we use an iterative relaxation method

that converges to a fixed point when $V(\vec{r})$ satisfies equation B.1. To simplify the computation we assume that our charge distribution will be radially symmetric, such as in the case of a circular domain, and we can thus transform the system into cylindrical coordinates (r, θ, z).

$$\frac{\epsilon(z)}{r}\frac{\partial V}{\partial r} + \frac{d\epsilon(z)}{dz}\frac{\partial V}{\partial z} + \epsilon(z)\frac{\partial^2 V}{\partial r^2} + \epsilon(z)\frac{\partial^2 V}{\partial z^2} = \epsilon(z)\kappa^2(z)V - \rho \tag{B.3}$$

We have assumed here that V and ρ are functions of only r and z, thus have dropped any differential terms that involve the azimuthal angle, θ. We treat z as being the direction normal to the membrane surface, which enables us to treat ϵ and κ as functions of z only.

B.1 Discretization Scheme

The next step is to derive an appropriate discretization scheme that we can use to find the relaxation condition. Consider a discrete $N_r \times N_z$ lattice in the (r, z) plane with box dimensions L_r and L_z. The lattice spacing would be given by $\Delta r = L_r/N_r$ and $\Delta z = L_z/N_z$. As shorthand we will use $V_{i,j} = V(i\Delta r, j\Delta z)$, and similarly for ρ, ϵ, and κ. For added computational simplicity we also assume that $\Delta r = \Delta z = \Delta$.

Using centered finite difference approximations for the derivatives in equation B.1 we can write the differential equation in the form,

$$V_{i,j} \Rightarrow \frac{B_{i,j}V_{i+1,j} + C_{i,j}V_{i-1,j} + D_{i,j}V_{i,j+1} + E_{i,j}V_{i,j-1} + \rho_{i,j}}{A_{i,j}} \tag{B.4}$$

$$\left\{ \begin{array}{l} A_{i,j} = \left(\epsilon_j \kappa_j^2 + \frac{4\epsilon_j}{\Delta^2}\right) \\ B_{i,j} = \left(\frac{(2i+1)\epsilon_j}{2i\Delta^2}\right) \\ C_{i,j} = \left(\frac{(2i-1)\epsilon_j}{2i\Delta^2}\right) \\ D_{i,j} = \left(\frac{\epsilon_{j+1}}{4\Delta^2} + \frac{\epsilon_j}{\Delta^2} - \frac{\epsilon_{j-1}}{4\Delta^2}\right) \\ E_{i,j} = \left(-\frac{\epsilon_{j+1}}{4\Delta^2} + \frac{\epsilon_j}{\Delta^2} + \frac{\epsilon_{j-1}}{4\Delta^2}\right) \end{array} \right\} \tag{B.5}$$

We use "$\cdot \Rightarrow \cdot$" to indicate a transformation, where a new value for the left hand side is computed using the current values of the right hand arguments. It is important to notice that $B_{i,j}$ and $C_{i,j}$ are not defined at $i = 0$. We resolve this by noting the symmetry $V(r, z) = V(-r, z)$ and expanding the $\partial V/\partial r$ term to a first order Taylor series at $r = 0$. It follows that for $i = 0$,

$$\left\{ \begin{array}{l} A_{0,j} = \left(\epsilon_j \kappa_j^2 + \frac{6\epsilon_j}{\Delta^2}\right) \\ B_{0,j} = \left(\frac{4\epsilon_j}{\Delta^2}\right) \\ C_{0,j} = 0 \end{array} \right\} \tag{B.6}$$

Using this set of equations we can evolve forward iteratively by replacing $V_{i,j}$ with the weighted sum of its surrounding lattice values using equation B.4. We do this until $V_{i,j}$ stops changing noticeably, at which point we know that V represents an approximate solution to equation B.1.

The dielectric constant, $\epsilon(z)$, the Debye screening length, $\kappa(z)$, and the charge density, $\rho(r, z)$, are defined prior to starting the simulation. For the geometry of the membrane we take the mid plane to be at $z = 0$. The hydrocarbon region has a half width of L_c and the headgroups are taken to have a thickness of L_h. This gives a total bilayer thickness of $2(L_c + L_h)$.

The dielectric constant is defined to be ϵ_w in the water on either side of the membrane and ϵ_c within the hydrocarbon chains. The dielectric in the headgroup region is smoothly interpolated between the two using a cubic spline [3, 5],

$$\frac{\epsilon(z)}{\epsilon_0} = \begin{cases} \epsilon_w & |z| > L_c + L_h \\ \frac{(\epsilon_c - \epsilon_w)}{4}\left[a(z)(a^2(z) - 3)\right] + \frac{\epsilon_c + \epsilon_w}{2} & L_c \leq |z| \leq L_c + L_h \\ \epsilon_c & |z| < L_c \end{cases} \tag{B.7}$$

$$a(z) = \left(\frac{2}{L_h}\right)|z| - \left(\frac{2L_c + L_h}{L_h}\right) \tag{B.8}$$

Here the affine transformation $a(z)$ is used as a shorthand for mapping the interval $[L_c, L_c + L_h] \to [-1, 1]$. The inverse Debye length, $\kappa(z)$, is defined in a similar piecewise manner, it is defined to be zero within the bilayer and some finite value κ in the surrounding medium,

$$\kappa(z) = \begin{cases} \kappa & |z| > L_c + L_h \\ 0 & |z| \leq L_c + L_h \end{cases} \tag{B.9}$$

The charge distribution, $\rho(r, z)$, is constructed to model a round domain of radius R_0 centered at $r = 0$. To emulate the effect of charge separation in the headgroup region of the bilayer we use two oppositely charged discs with charge density ρ [6, 1, 2]. The negatively charged disc is placed at the headgroup/water interface and the positively charged disc at the acyl chain/headgroup interface. The charge distribution is,

$$\rho(r, z) = \begin{cases} 0 & r > R_0 \\ \rho\delta(|z| - L_c) - \rho\delta(|z| - (L_c + L_h)) & r \leq R_0 \end{cases} \tag{B.10}$$

This arrangement ensures that we will have a higher electrostatic potential within the membrane than in the surrounding medium, consistent with experimental observations [4, 8]. For the boundary conditions of the simulation we hold $V_{i,j} = 0$ at the edges of the box. The full

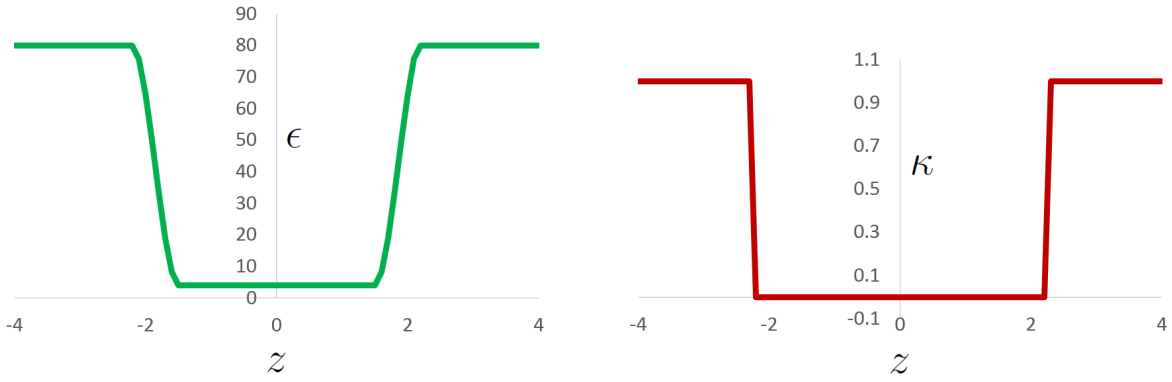


Figure B.1: Functions used for $\epsilon(z)$ and $\kappa(z)$ in the Debye-Huckel equation. The dielectric is varied continuously from ϵ_w in the water to ϵ_c in the hydrocarbon. The Debye length is discontinuous, being finite in solution and infinite in the bilayer (no salt screening).

set up of the simulation is shown in figures B.1, B.2.

B.2 Results

This modeling produces a full three dimensional electrostatic potential map of a round domain in a bilayer submerged in an ionic aqueous environment. Using these data we can compute the electric fields in and around the bilayer and answer any number of questions about how they vary in space. The two questions most immediately relevant to our work were,

1. Does the electric field within the bilayer, particularly the headgroup region, decay exponentially or as a power law?

2. If it does decay exponentially, how long is the decay length?

In addition we can ask how the answers to these questions depend on the size of the domain, R_0, and the Debye length in the surrounding medium, $1/\kappa$. The parameters used in the simulation are shown in table B.1. Those with multiple values indicate that a series of

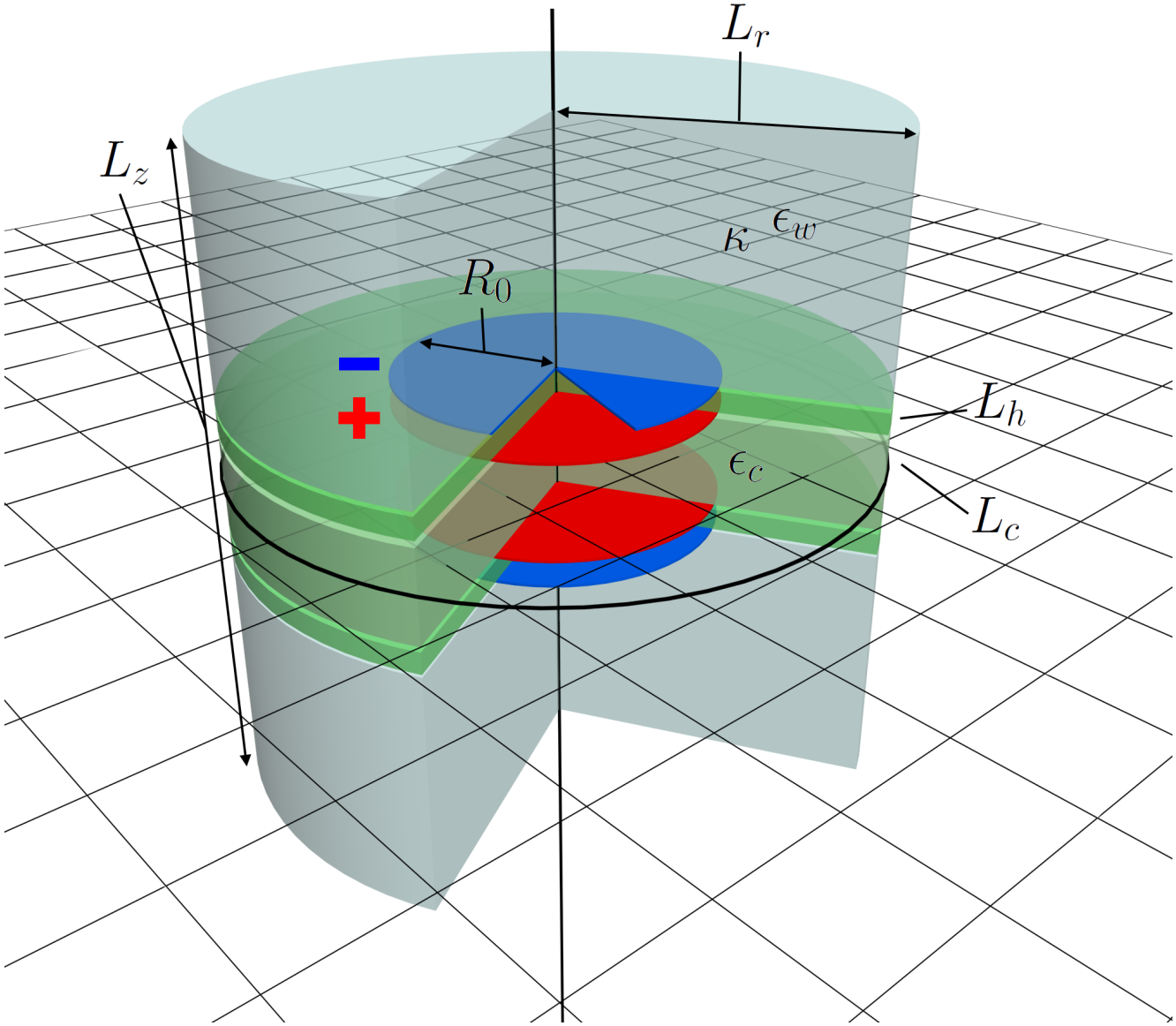


Figure B.2: Geometric layout of the charge distribution. The headgroups of the bilayer are modeled as circular capacitor plates of opposite charge. The parameters used for each dimension are given in table B.1.

parameter	value	unit
L_r	20	nm
L_z	40	nm
N_r	1000	N/A
N_z	2000	N/A
L_c	1.5	nm
L_h	0.7	nm
ϵ_w	80	N/A
ϵ_c	4	N/A
R_0	$4, 8, 12$	nm
κ	$0.0, 0.2, 1.0, \infty$	nm^{-1}
ρ	1	e^-/nm^3

Table B.1: Parameters used for the solution to the Debye-Huckel equation.

values was tested. The values of κ shown correspond (from left to right) to no salt, 4mM, 94mM, and ∞mM monovalent salt concentrations. The interpretation of ∞mM salt is that the bilayer is flanked by grounded conducting planes.

The charge density, ρ, is a degree of freedom within the simulation and the value $\rho = 1$ is chosen for convenience. After the simulation has run to completion we can constrain ρ by rescaling the voltage to match experimental values. If we define $\Delta V^{(\mathrm{sim})} \equiv V(0, L_c) - V(0, L_c + L_h)$ then the new value of ρ is,

$$\rho^{(\mathrm{new})} = \rho^{(\mathrm{old})} \frac{\Delta V^{(\mathrm{exp})}}{\Delta V^{(\mathrm{sim})}} \tag{B.11}$$

where the value of $\Delta V^{(\mathrm{exp})} \approx 300\mathrm{mV}$ [7]. The reason we can do this post-hoc correction is because the Debye-Huckel equation is linear with respect to scaling of ρ.

The results of this simulation for $R_0 = 8\mathrm{nm}$ and $\kappa = 0.2\mathrm{nm}^{-1}$ are shown in figure B.3. The electrostatic potential outside of the membrane is close to zero, as the salt prohibits the electric fields from penetrating too far into the medium. Within the headgroup region we have strong static electric fields that are normal to the membrane (pointing away from the positively charged interior). Our interest is in the fringe fields at the edge of the domain. A

test lipid will experience these fields as it approaches the domain edge. To measure this we calculate the voltage in the center of the headgroup region,

$$V_h(x) \equiv V\left(R_0 + x, L_c + L_h/2\right) \tag{B.12}$$

and express it as a function of radial distance from the domain edge, x. By plotting $\log V_h(x)$ we find a linear plot with negative slope, indicating an exponential decay of the electrostatic potential as we move away from the edge of the domain. Thus the answer to question (1) is that the electric fields do decay exponentially.

The answer to question (2) can be ascertained by taking the slope of the $\log V_h(x)$ curve, which gives us the inverse decay length of the electric field,

$$k = -\left(\frac{d[\log V_h]}{dx}\right) \tag{B.13}$$

From the plot in figure B.4, we can see that there is only a limited region where $\log V_h$ appears linear. At the edge of the domain there is discontinuous step in the charge density which gives a steep change in the voltage, towards the right side, we get finite size artifacts because $V(L_r, z) = 0$ is fixed by the boundary conditions. We measure k far from the domain edge and far from the edge of the simulation box in the linear region.

The $\log V_h(x)$ curve for each of the salt concentrations in table B.1 is also shown, with their respective values of k^{-1}. Notice that even for 0mM salt concentration we still get an exponentially decaying V_h, with a decay length of $k^{-1} \approx 3.7$nm. Adding salt decreases this length, with a minimum of $k^{-1} \approx 1$nm in the extreme case of ∞ salt. The same calculation was done for different domain sizes, R_0, with identical results.

The observation that the fields still decay exponentially even in the absence of salt was a bit surprising. It is known that in lipid monolayer studies the electrostatic interaction from lipid dipoles is long range (varies as a power law with distance) and can affect domain

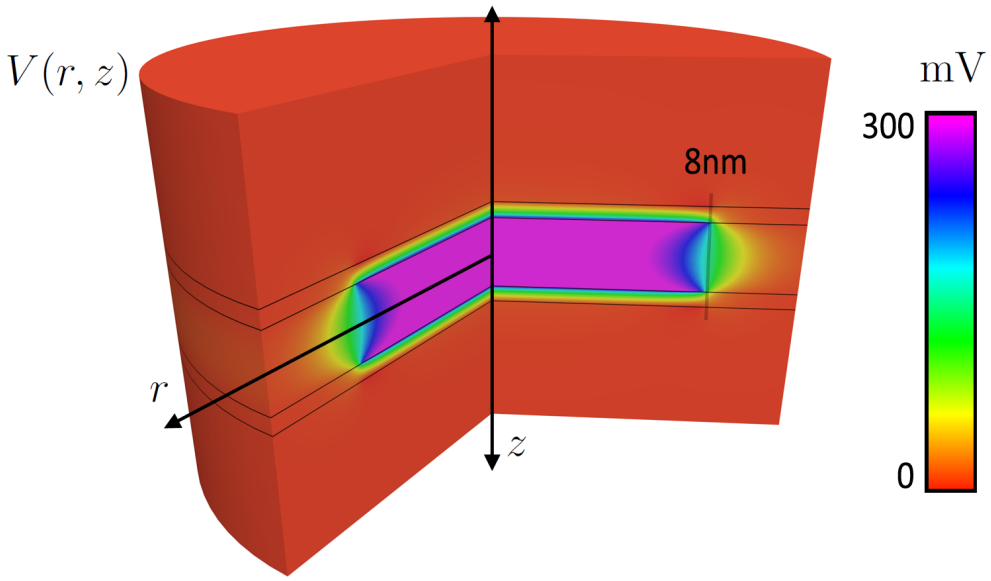


Figure B.3: Cross section of electrostatic potential in the interior of a circular domain. Charge densities were re-scaled to produce a 300mV potential difference between inside and outside of the membrane according to equation B.11.

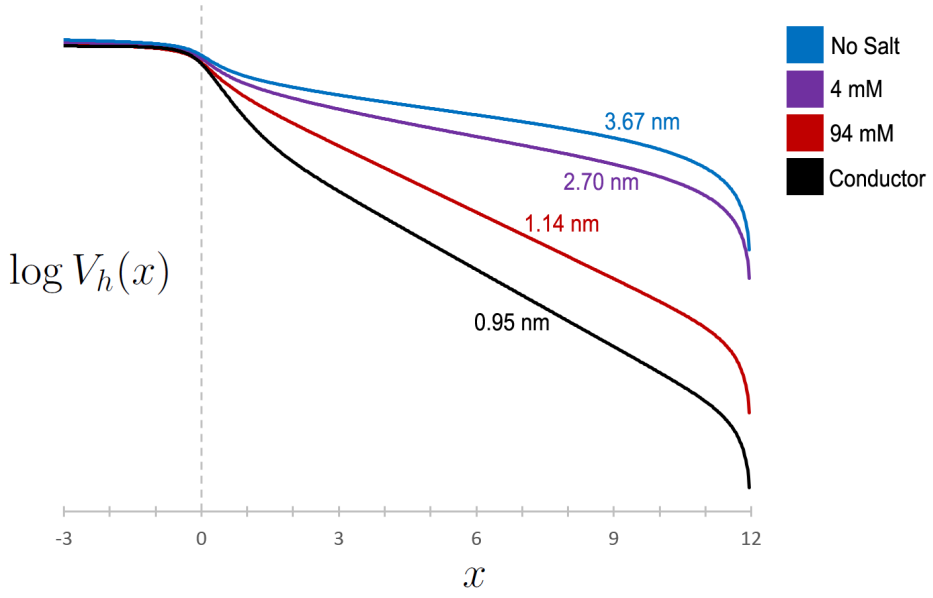


Figure B.4: Log plot of V_h for different Debye lengths showing the electrostatic potential in the headgroup region at the edge of the domain. Linear regions of the plot indicate exponential decays with decay lengths shown in the respective colors.

morphology on optical scales. The short decay length in bilayers can be attributed to the symmetry of the bilayer. Placing two oppositely oriented dipole sheets within close proximity will self-screen and produce a decay length on the order of the spacing between the sheets (in our case $2L_c = 3$nm).

The results of this modeling are that the electrostatic interaction between two adjacent lipids decays exponentially within the headgroup region of the bilayer. This decay length depends both on thickness of the bilayer and the salt concentrations in the surrounding medium. With no salt we find that the field decay length is $k^{-1} = 3.7$nm and with infinite salt concentration (a perfectly conducting medium), the decay length is only $k^{-1} = 1$nm. In both cases this decay length is too small to explain optical structures (such as modulated phases) observed on GUVs.

Bibliography

[1] A. Demchenko and S. Yesylevskyy. Nanoscopic description of biomembrane electrostatics: Results of molecular dynamics simulations and fluorescence probing. *Chem. Phys. Lipids*, 160:63–84, 2009.

[2] A. Imperio and L. Reatto. Microphase separation in two-dimensional systems with competing interactions. *J. Chem. Phys.*, 124:164712, 2006.

[3] H. Nymeyer and H. Zhou. A method to determine dielectric constants in nonhomogeneous systems: Application to biological membranes. *Biophysical Journal*, 94:1185–1193, 2008.

[4] E. Pohl. *Advances in Planar Lipid Bilayers and Liposomes*, volume 1. Elsevier Inc., 2005.

[5] A. Raudino and D. Mauzerall. Dielectric properties of the polar head group region of zwitterionic lipid bilayers. *Biophysical Journal*, 50:441–449, 1986.

[6] A. Travesset. Effect of dipolar moments in domain sizes of lipid bilayers and monolayers. *J. Chem. Phys.*, 125:084905, 2006.

[7] L. Wang. Measurements and implications of the membrane dipole potential. *Annu. Rev. Biochem.*, 81:615–635, 2012.

[8] L. Wang, P. Bose, and F. Sigworth. Using cryo-em to measure the dipole potential of a lipid membrane. *PNAS*, 103:18528–18533, 2006.

Appendix C

Unbiased Spherical Lattice

To construct a spherical lattice we started by uniformly dispersing N points in the three dimensional volume $[-1,1]^3 \subset \mathbb{R}^3$. This produces a set of indexed points, $\{\vec{v}_i\}$, in the unit cube centered at the origin. We then project these points onto the unit sphere using the transformation,

$$\vec{v}_i \Rightarrow \frac{\vec{v}_i}{\|\vec{v}_i\|} \tag{C.1}$$

Where $\|\cdot\|$ is the Euclidean norm. This produces a semi-uniformly distributed set of points on the unit sphere (the density will tend to be higher towards the corners of the initial cube). Our goal is to produce a collection of points on the unit sphere that form a mono-dispersed regular array.

To force the points into a regular array we make use of a simplified charge relaxation approach. Each vertex is given a repulsive interaction with every other point. For our relaxation we use the following force law,

$$F_{ij} = \frac{\vec{v}_i - \vec{v}_j}{\|\vec{v}_i - \vec{v}_j\|} \tag{C.2}$$

We then assume damped dynamics, where the displacement of a given point is proportional to the total force on that point,

$$\Delta\vec{l_i} = \alpha \sum_j F_{ij} \tag{C.3}$$

Where α is chosen so that the displacements remain small within a single relaxation step (usually $\alpha = 0.001$). We also must be careful to ensure that the point does not leave the unit sphere, as this displacement may be in any direction. The new position of $\vec{v_i}$ after a single relaxation step is given by,

$$\vec{v_i} \Rightarrow \frac{\vec{v_i} + \Delta\vec{l_i}}{\left\|\vec{v_i} + \Delta\vec{l_i}\right\|} \tag{C.4}$$

We repeat this relaxation procedure until all of the vertices remain fixed under subsequent iterations ($\Delta\vec{l_i} = 0$). The resultant set of points is mono-dispersed on the surface of the unit sphere, as this minimizes the frustration with respect to F_{ij}.

C.1 Lattice Connectivity

The next step is to define the connectivity of the lattice. For each vertex we define a list of indices of potential neighboring vertices, n_i,

$$n_i = \{j \mid \|\vec{v_i} - \vec{v_j}\| < \beta r_{min}\} \tag{C.5}$$

Where $r_{min} = \min\{\|\vec{v_i} - \vec{v_j}\|, \forall i, j\}$ is the minimum distance between any two vertices. Define b_i to be the number of elements in n_i. The multiplier β is empirically determined to find a good starting place for defining the connectivity. We start with $\beta = 1$ and slowly increase it until $4 \leq b_i \leq 8$, $\forall i$. We find that this condition is met for $\beta \approx 1.5$. This ensures

that every vertex has between 4 and 8 neighbors. There will always be defects in this way of constructing a neighbor array, but we will classify and repair these later.

C.2 Ordering the Neighbor Arrays

The next important step is to order this list. We want the neighbors of a given vertex to form a counter clockwise loop around the parent vertex. This is done iteratively by comparing the cross products of pairs of neighboring vertices. Let $n_i[0]$ be the first element of the neighbor array. We define the set of counter clockwise candidates, n^+, to be the set of values in n_i that satisfy,

$$n^+ = \left\{ n_i[j] \mid \left(\frac{\vec{v}_i - \vec{v}_{n_i[0]}}{\left\|\vec{v}_i - \vec{v}_{n_i[0]}\right\|} \times \frac{\vec{v}_i - \vec{v}_{n_i[j]}}{\left\|\vec{v}_i - \vec{v}_{n_i[j]}\right\|} \right) \cdot \frac{\vec{v}_i}{\|\vec{v}_i\|} > 0 \right\} \tag{C.6}$$

One of the elements of n^+ is the counter clock-wise neighbor of $n_i[0]$. To determine which element this is we find the element of n^+ which is closest in direction to $n_i[0]$,

$$j^* = \underset{j}{\text{argmax}} \left\{ \left(\frac{\vec{v}_i - \vec{v}_{n_i[0]}}{\left\|\vec{v}_i - \vec{v}_{n_i[0]}\right\|} \right) \cdot \left(\frac{\vec{v}_i - \vec{v}_{n_i[j]}}{\left\|\vec{v}_i - \vec{v}_{n_i[j]}\right\|} \right), n_i[j] \in n^+ \right\} \tag{C.7}$$

The value $n_i[j^*]$ is the index of the counter clock-wise neighbor of $n_i[0]$. Thus we swap $n_i[1] \Leftrightarrow n_i[j^*]$. We repeat this process for $n_i[1], ..., n_i[b_i - 1]$ to produce a complete counter clock-wise loop around $\vec{v}_i$. In practice this routine is automated into a function call, so that at any point during the construction of the lattice the neighbor arrays can be ordered.

C.3 Repairing Hatches and Holes

As mentioned before this process produces a "rough draft" of the final lattice and still contains many defects. The two types of defects that must be addressed are,

normal	$\{v_1, v_3\} \subset n_0$ and $\{v_2, v_3, v_0\} \subset n_1$
normal	$\{v_1, v_2, v_3\} \subset n_0$ and $\{v_2, v_0\} \subset n_1$
hatch	$\{v_1, v_2, v_3\} \subset n_0$ and $\{v_2, v_3, v_0\} \subset n_1$
hole	$\{v_1, v_3\} \subset n_0$ and $\{v_2, v_0\} \subset n_1$

Table C.1: Various expected neighbor arrays for normal and defective connections.

1. **Hatches**, which occur when two adjacent vertices form too many connections and produce intersecting edges.

2. **Holes**, which occur when two adjacent vertices form too few connections, leaving an open hole in the surface.

To automate the process of identifying and repairing these defects we work with quads of adjacent vertices and search for key sequences in their neighbor arrays that indicate an anomalous connection. Consider the vertex quad v_0, v_1, v_2, v_3 with neighbor arrays n_0, n_1, n_2, n_3. The four possible sequences that we look for are shown in table C.1. Once a defect is found, repairing it is straightforward. In the case of a hatch, we remove v_2 from n_0 and v_0 from n_2. For a hole we insert v_2 into n_0 and insert v_0 into n_2. The neighbor arrays are re-ordered after insertions or deletions to maintain a counter clock-wise orientation. These defects before and after repair are shown in figure C.1.

There is a subtle problem with this way of repairing defects in the lattice: we have assumed that defects are local to the immediate quad about a given vertex. This need not be the case, since a hatch may connect vertices several neighbors away and holes may be larger than a single quad. The hatch-repair algorithm fails spectacularly when connections are beyond nearest neighbors, but the hole-repair algorithm does work in the general case (it fills in the hole one edge at a time with each iteration).

We use the fact that the hole-repair algorithm works in general to avoid this problem. With a conservative choice of β in our initial connection sweep we ensure that hatches will be local, even if that means holes are more numerous. Once all defects have been repaired

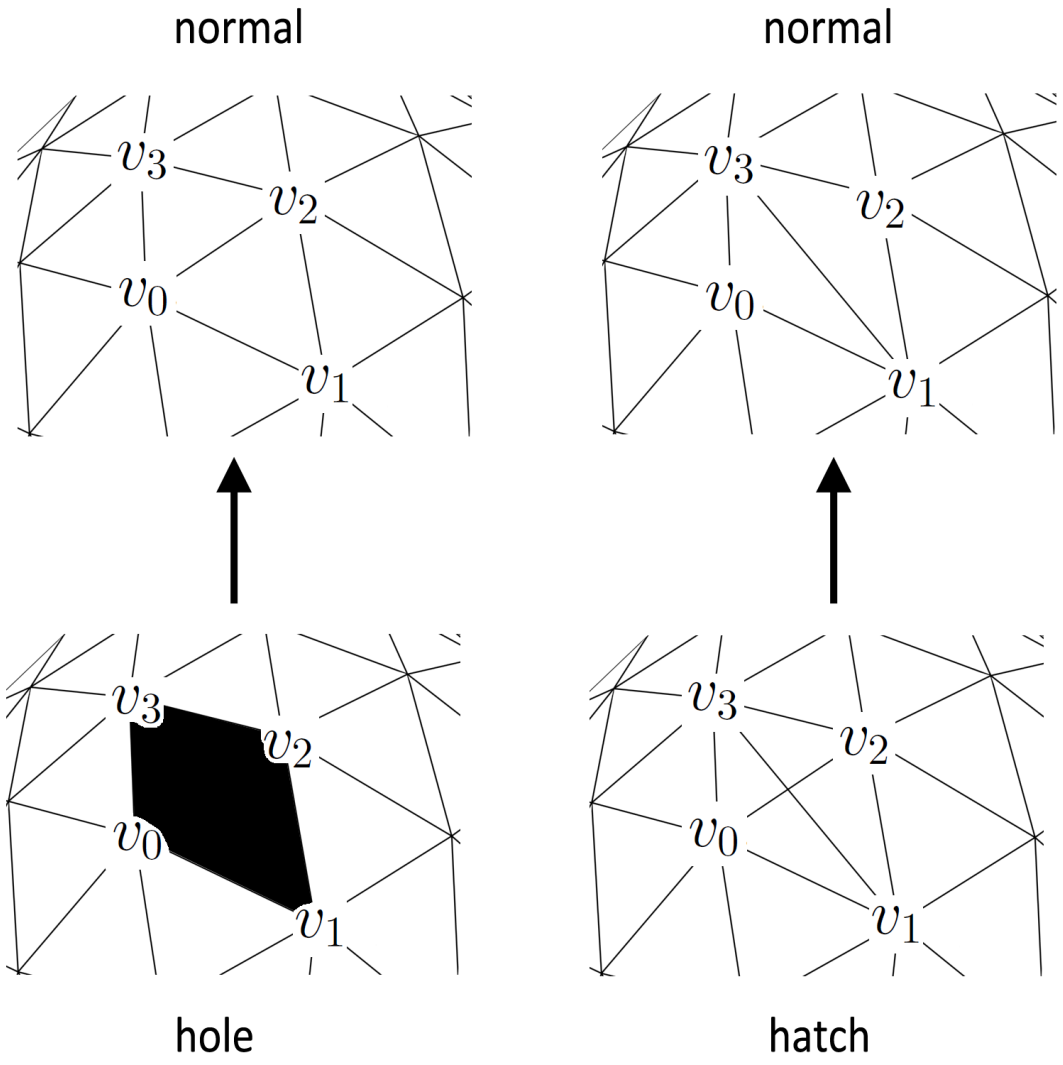


Figure C.1: Two varieties of defects observed in connected lattice. These defects are identified and repaired using the neighbor arrays shown in table C.1.

and the neighbor arrays are all properly ordered we have a completed orientable surface. A summary of these steps is shown in figure C.2.

C.4 Triangulating Arbitrary Surfaces

The algorithm outlined in this section is very general and so long as a suitable projection and force law can be defined it can be used to produce a triangulated lattice of an arbitrary non-intersecting surface, S. The F_{ij} defined in equation C.2 is not directly applicable in the general case; it worked for the sphere because of the high degree of symmetry. For the general case a more suitable force law is be given by,

$$F_{ij} = \begin{cases} \infty & \|\vec{v}_i - \vec{v}_j\| < d_0 \\ 1/(\|\vec{v}_i - \vec{v}_j\| - d_0)^2 & \|\vec{v}_i - \vec{v}_j\| \geq d_0 \end{cases} \tag{C.8}$$

Where d_0 is a minimum inter-vertex distance defined as $\sqrt{A_0/(\pi N_v)}$, where A_0 is the area of the surface and N_v is the number of vertices. In practice we would set d_0 slightly below this value to allow for some flexibility in the steric interaction of adjacent vertices.

A projection function is necessary to constrain points to move only within the surface. For an arbitrary point in space $\vec{v}$, define the closest point on the surface as,

$$\vec{s}* = \underset{\vec{s} \in S}{\operatorname{argmin}} \|\vec{v} - \vec{s}\| \tag{C.9}$$

This closest point is defines the projection onto the surface, $\vec{v} \Rightarrow \vec{s}*$. Iterative relaxation is performed, and determining the connectivity of the lattice works exactly as outlined in the sections C.1, C.2, and C.3. This produces an unbiased triangulation of an arbitrary surface.

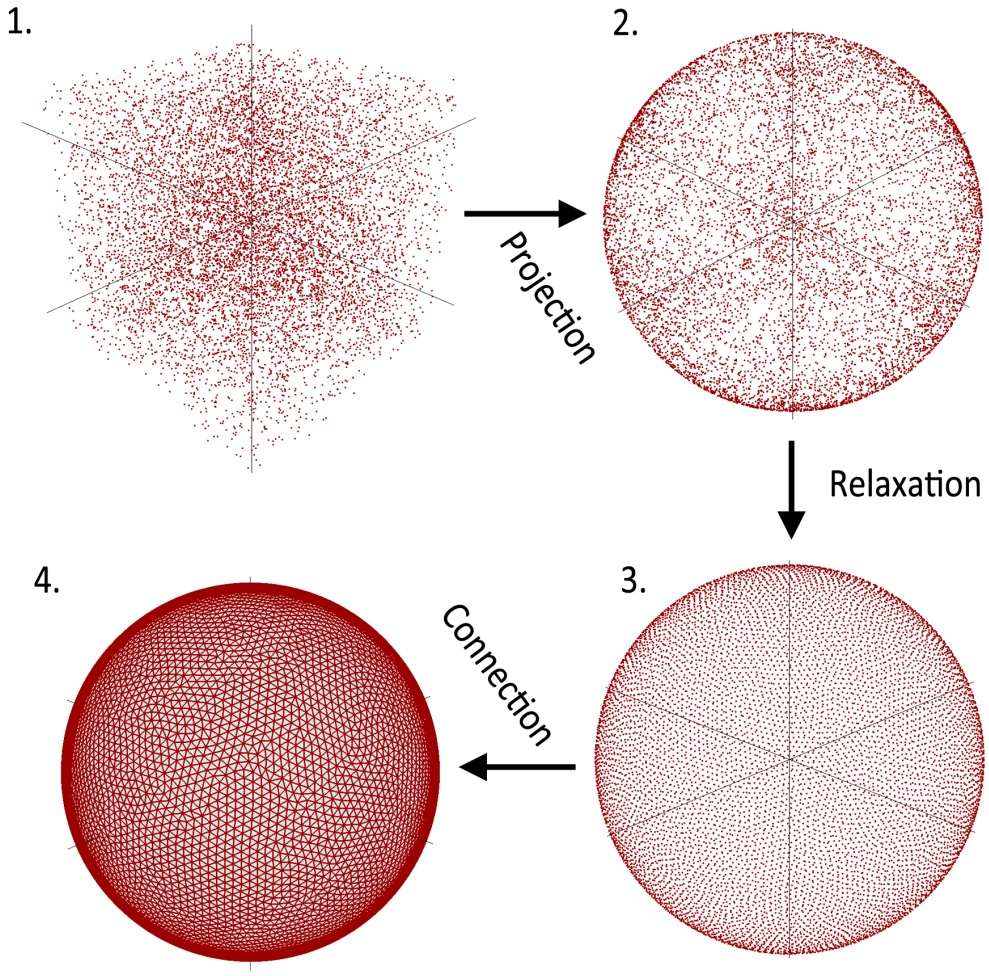


Figure C.2: Summary of steps used to generate an unbiased lattice.

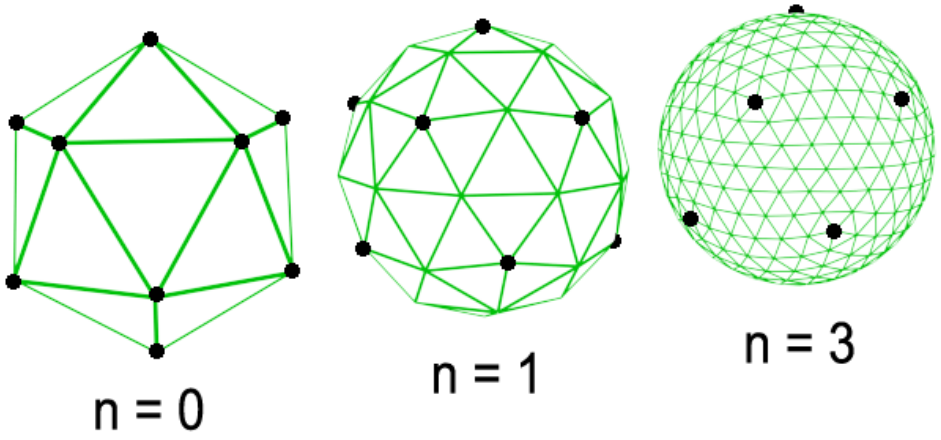


Figure C.3: Example of geodesic subdivision, a deterministic and iterative way to triangulate a sphere. Successive iterations are shown from left to right with the original icosahedron vertices highlighted in black.

C.5 Geodesic Subdivision

There do exist deterministic ways to triangulate a sphere that were also tried, such as geodesic subdivision. This produces a lattice that is almost perfectly hexagonal, except for twelve special points (which correspond to an inscribed icosahedron) that have five neighbors. These defects biased our simulation towards morphologies/shapes that mirrored this icosahedral symmetry. The algorithm of geodesic subdivision starts with an icosahedron. Each face is cut along the midpoints of its edges to create an inscribed triangle whose vertices are projected onto the unit sphere. This process can be repeated iteratively producing the series of spherical lattices shown in figure C.3.

The construction of a spherical lattice using the iterative relaxation routine outlined above, instead of a geodesic subdivision, is a way of defeating this bias. Instead of creating a perfectly hexagonal tiling (with icosahedral defects), we create a highly entropic lattice in which the connectivity is approximately hexagonal and defects are randomly distributed across the surface.

Appendix D

Phosphorescence Lifetime Model

Phosphorescence is one of the possible pathways that a photon may take after being absorbed by a photoactive molecule. Fluorescent molecules absorb and re-emit photons (at a longer wavelength) very quickly, on the order of nanoseconds. This makes fluorescent probes ideal for studying static properties of the membrane. On the other hand, phosphorescence involves forbidden transitions to and from a triplet state, which may last many microseconds or even far longer [2]. This gives phosphorescent probes access to an interesting time scale, where the dynamics of individual lipids or small domains may be resolvable.

Here we present a model that may find utility in matching phosphorescence lifetime measurements to the underlying geometry of phase domains under certain conditions. This model was never put into use due to technical limitations of the experimental system, but is outlined in detail for future reference when model assumptions and experimental parameters can be met.

D.1 Model Outline

Phosphorescence lifetime has been found to be very sensitive to the immediate environment of the molecule. Thus it may be that the decay lifetime for a phosphorescent probe is different in different bilayer phases. Let these two lifetimes be given by τ_d and τ_o for the L_d and L_o phase respectively. Suppose a test probe is set free on a phase separated membrane with a fixed domain geometry. This probe will diffuse in and out of the domain as it explores the membrane and eventually emit a photon. The effective phosphorescence decay curve that would be measured for an ensemble of such test probes would be modulated by the path of the probe and how much time it spent in each phase.

To formalize this consider a diffusion path $s(t)$ in the membrane plane and an ensemble of probes that follow this path. The probability density $p_s(t)$ that it will emit a photon at time t is given by,

$$\frac{dp_s}{dt} = -\frac{p_s}{\tau(s(t))} \Rightarrow p_s(t) \propto \exp\left[-\int_0^t \frac{dt}{\tau(s(t))}\right] \tag{D.1}$$

$$\tau(r) = \begin{cases} \tau_d & r \in L_d \\ \tau_o & r \in L_o \end{cases} \tag{D.2}$$

Note that for a single environment we would have $\tau(s(t)) =$ constant, Which gives a pure exponential decay with a single time constant as expected. Equation D.1 only captures the effect of a single diffusion path. In order to get the expected decay curve from a large number of probes, all excited at once and diffusing along different paths, we must average $p_s(t)$ over all possible diffusion paths. The expected decay curve $\langle p \rangle (t)$ is given by,

$$\langle p \rangle (t) = \sum_s p_s(t) \mathcal{D}s \tag{D.3}$$

The weighting $\mathcal{D}s$ is a probability measure for the diffusion path s. To perform this sum it is best to approximate the space of all diffusion paths in such a way that $\mathcal{D}s$ is roughly constant. A pragmatic choice is to use the set of all random walks, which should all occur with equal amplitude and are very easy to simulate.

To perform the sum in equation D.3 a Monte Carlo simulation is used that assumes a fixed domain geometry, a fixed time step Δt, and a diffusion constant of D_d and D_o for the L_d and L_o phases respectively. Probes are placed randomly on the plane and are allowed to random walk, with step sizes Δr given by,

$$\Delta r = \begin{cases} \sqrt{4D_d\Delta t} & r \in L_d \\ \sqrt{4D_o\Delta t} & r \in L_o \end{cases} \tag{D.4}$$

To simulate the effects of a partition coefficient K_p (ratio of concentrations in L_d and L_o), any random step that crosses a phase boundary is accepted or rejected with probability $K_p/(1+K_p)$ for $L_o \to L_d$ and probability $1/(1+K_p)$ for $L_d \to L_o$. Averaging the resultant decay curves over a large number of trials (10,000 was found to be adequate) converges to a good approximation of $\langle p \rangle (t)$. This is the expected decay curve we would obtain from a bulk phosphorescence lifetime measurement on a phase-separated membrane.

D.2 Relating Decay Curve to Geometry

The next step is relating the characteristics of $\langle p \rangle (t)$ to geometric parameters of the phase domains. To do this some assumptions about the mathematical form of $\langle p \rangle (t)$ have to be made,

$$\langle p \rangle (t) = A_d e^{-t/\tau_d} + A_m e^{-t/\tau_m} + A_o e^{-t/\tau_o} \tag{D.5}$$

The terms with coefficient A_d and A_o correspond to the bulk contributions of probes that

do not diffuse in and out of domains, but instead remain in their starting environment for the duration of their lifetime. The center term A_m corresponds to a simplified mixing component with an effective phosphorescence lifetime of τ_m. The curve is assumed to be normalized ($\tau_d A_d + \tau_m A_m + \tau_o A_o = 1$). Right away this model provides some intuitive predictions for extreme cases:

If the domain dimensions R_{domain} are much larger than the distance a probe would diffuse before emitting a photon, $R_{\text{domain}} >> \max\left\{\sqrt{4D_d\tau_d}, \sqrt{4D_o\tau_o}\right\}$, then we would expect the mixing term to be negligible. This leads to a relationship between the coefficients A_d and A_o, the L_o area fraction P, and the partition coefficient of the probe K_p.

$$A_d = \frac{(1-P)K_p}{\tau_d[(1-P)K_p + P]} \qquad A_o = \frac{P}{\tau_o[(1-P)K_p + P]} \tag{D.6}$$

Notice that there is no dependence on R_{domain}, so this gives no geometric information, but can be used to estimate P and K_p if the phase domains are known to be macroscopic.

The opposite extreme is to assumes that the phase domain size is much smaller than the distance a probe would diffuse before emitting, $R_{\text{domain}} << \max\left\{\sqrt{4D_d\tau_d}, \sqrt{4D_o\tau_o}\right\}$. In this case all probes experience both environments in proportion dependent on P and K_p. This leaves only the mixing term with a single lifetime.

$$\frac{1}{\tau_m} = \frac{(1-P)K_p}{\tau_d[(1-P)K_p + P]} + \frac{P}{\tau_o[(1-P)K_p + P]} \tag{D.7}$$

Again this expression has no dependence on R_{domain}, so we cannot make any geometric estimates from this measurement. This case contains less information than the large domain case, as the determination of the single parameter τ_m cannot yield the values of P and K_p simultaneously.

The ability of this model to probe domain geometry lies in a sweet-spot of domain sizes, where the number of probes that experience a single environment (either L_o or L_d) and the

number of probes that mix between the two are of roughly equal proportion. An estimate of this size scale can be estimated by the distance a probe diffuses before emitting. With a phosphorescent lifetime of 50μs and a diffusion constant of 10^{-8} cm^2/s, this gives a size scale of about 10nm. A Monte Carlo simulation of this case is shown in figure D.1. The probes that mix between the two environments (cyan and yellow) form a halo around the domain boundaries, which makes the value of A_m a first order approximation of the total length of the domain boundary L.

$$\tau_m A_m = \frac{L}{A}\frac{(K_p l_d + l_o)}{K_p + 1} + O(l_d^2, l_o^2) \tag{D.8}$$

$$\left\{ \begin{array}{l} l_d = \sqrt{4 D_d \tau_o} \\ l_o = \sqrt{4 D_o \tau_d} \end{array} \right\} \tag{D.9}$$

where A is the total area of membrane. The identification of A_m with the area of this mixing halo can be used to estimate R_{domain} when a domain geometry is chosen. As a simple example case, consider circular domains of radius R_{domain}. The expression for R_{domain} to first order in l_d and l_o is,

$$R_{\text{domain}} \approx \frac{2P(K_p l_d + l_o)}{\tau_m A_m (K_p + 1)} \tag{D.10}$$

This is a crude approximation, but may be good enough for an order of magnitude estimate of the size of small phase domains, or perhaps to track a trend in domains size under some perturbation, such as temperature or the addition of protein. Note that an exact expression for circular and stripe domain cases can be worked out by expanding equation D.8 to second order.

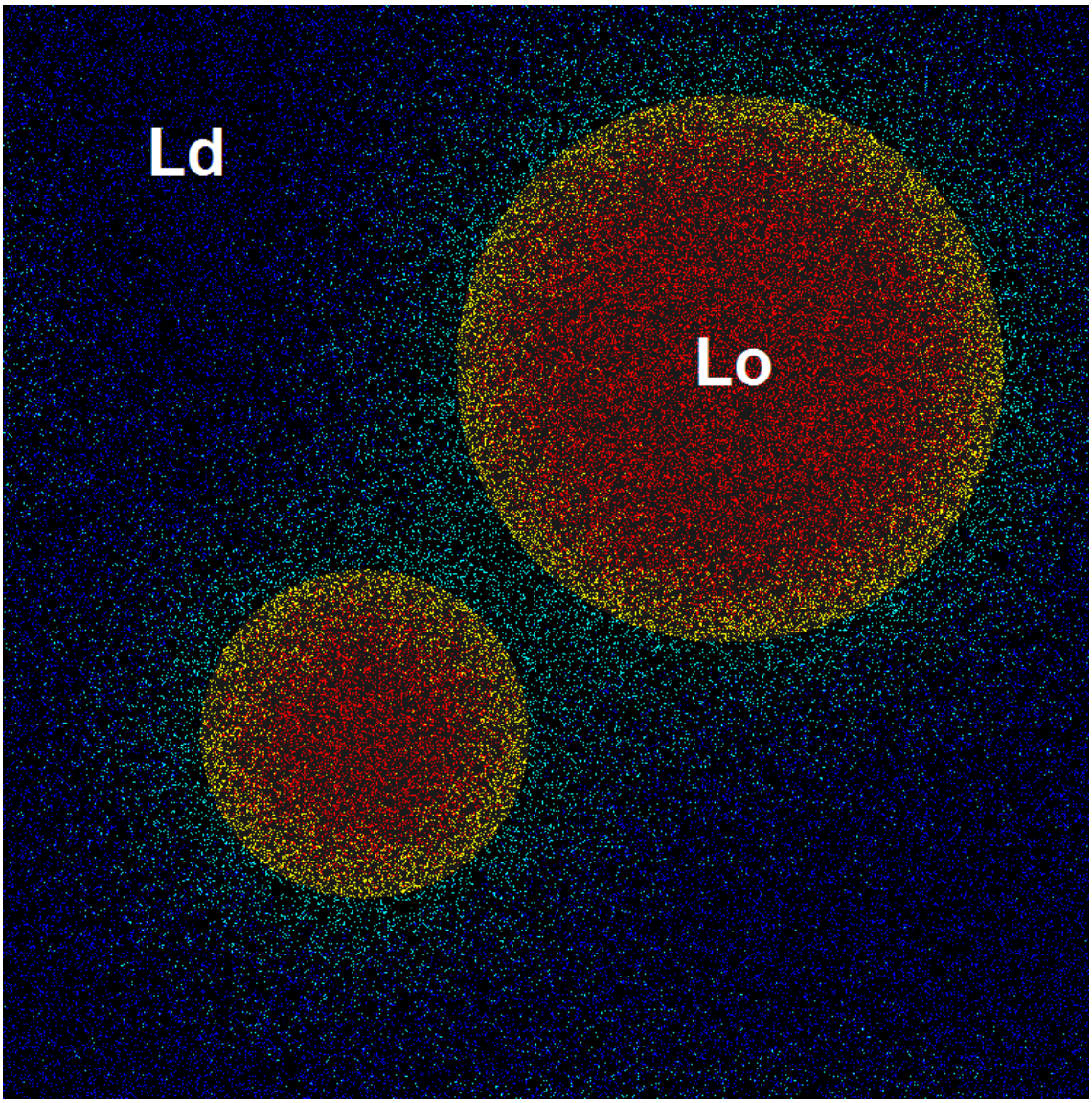


Figure D.1: Simulated phosphorescent probes on an L_d membrane with circular L_o domains. Each colored point represents an individual phosphorescent probe: (blue) probes that started in L_d and never crossed a domain boundary, (red) probes that started in L_o and never crossed a domain boundary, (cyan) probes that started in L_d, but diffused across a domain boundary, (yellow) probes that started in L_o, but diffused across a domain boundary.

D.3 Analyzing Measured Decay Curves

The model presented so far can take a given domain geometry and set of probe characteristics and be used to simulate the decay curve that would result from a bulk measurement. In practice we will be doing the opposite, taking a bulk measured decay curve and resolving the phase domain geometry from that. This problem comes down to fitting the measured decay curve to the form presented in equation D.5.

Accurately fitting multi-exponential functions to data is a notoriously difficult task [5]. We can make this fitting problem somewhat easier by noting that two of the lifetimes, τ_d and τ_o, can be measured independently by preparing one-phase mixtures and fitting to a single exponential. The coefficients A_d, A_o, and A_m, must also satisfy the normalization conditions, further reducing the number of free parameters. A non-linear χ^2 fit can be used to determine the values of A_m and τ_m. The values of P and K_p must be measured by other means to be used in equation D.10.

A novel technique that may also prove useful in fitting equation D.5 to data may be the use of phasor analysis [1, 3, 4]. The appeal of this method is that it decouples the determination of τ_m and A_m, linearizes the fitting calculation for A_m, and enables us to use more information about the system (such as P and K_p) to better constrain the solution. A graphical illustration of the phasor solution is shown in figure D.2. The phasor of a normalized decay curve is the real and imaginary part of its Fourier transform at a chosen frequency ω (this is a free parameter, but it is best to choose $\omega \approx 1/\tau$). The phasors for a pure decay curve all lie along a semicircle in the complex plane, termed the 'universal circle.' We can plot the positions of the one-phase decay curves, $\vec{v_o}$ and $\vec{v_d}$, on this circle (blue and red markers). Any linear combination of the two decay curves will lie along the line connecting $\vec{v}_o$ and $\vec{v}_d$.

Using the values of P and K_p we can use the expressions for the bulk phase separated

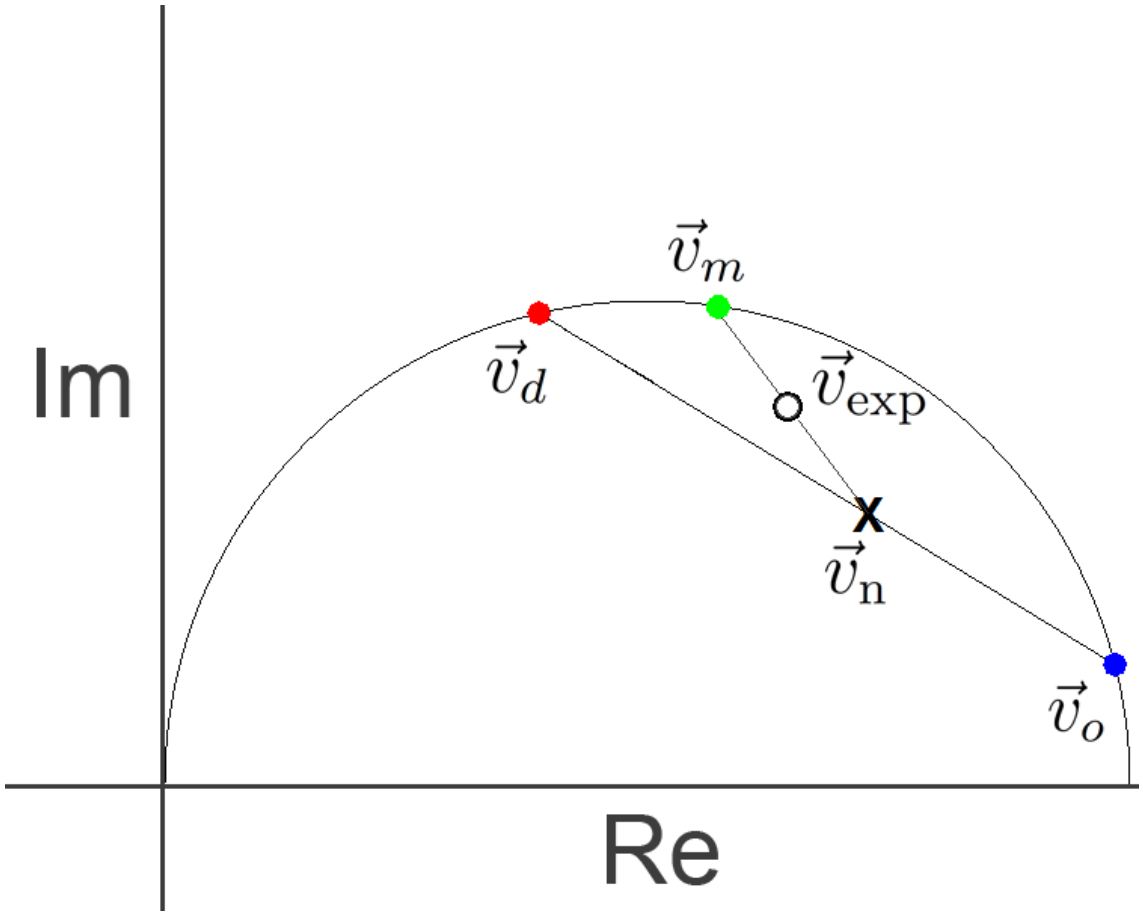


Figure D.2: Visual representation of phasor analysis. The two pure phase phasors, $\vec{v}_d$ and $\vec{v}_o$, are plotted along the universal circle (red and blue circles). Then the values of P and K_p are used to locate the expected 'no mixing' phasor $\vec{v}_n$ along the adjoining line segment (black **X**). The phasor of the experimental data $\vec{v}_{\exp}$ is plotted (open circle) and a mixing phasor $\vec{v}_m$ (green circle) is extrapolated by drawing a line through $\vec{v}_n$ and $\vec{v}_{\exp}$ to intersect the universal circle. The three basis vectors $\vec{v}_d$, $\vec{v}_o$, and $\vec{v}_m$ are used to decompose $\vec{v}_{\exp}$ and determine the values of A_d,A_o, and A_m.

system (equation D.6) to plot the expected phasor $\vec{v}_{\mathrm{n}}$ on this line segment (black **X** marker) if no mixing were present. We then plot the phasor of the measured decay curve $\vec{v}_{\mathrm{exp}}$ (open circle). If there is a significant proportion of probes mixing between the two phases then $\vec{v}_{\mathrm{exp}}$ would be expected to lie above the line segment connecting $\vec{v}_o$ and $\vec{v}_d$. The mixing lifetime τ_m is then extrapolated by drawing a line through $\vec{v}_{\mathrm{n}}$ and $\vec{v}_{\mathrm{exp}}$ and determining where it intersects the universal circle $\vec{v}_m$ (green marker). These three reference points, $\vec{v}_o$, $\vec{v}_d$, and $\vec{v}_m$, together with the normalization condition, can be used to decompose $\vec{v}_{\mathrm{exp}}$ into its three components by solving a system of linear equations.

$$\left\{ \begin{array}{c} A_d\vec{v}_d + A_m\vec{v}_m + A_o\vec{v}_o = \vec{v}_{\mathrm{exp}} \\ A_d\tau_d + A_m\tau_m + A_o\tau_o = 1 \end{array} \right\} \tag{D.11}$$

Solving these equations gives us all of the information that we need to apply equation D.10 or any other expression for the assumed domain geometry. The disadvantages of this phosphorescence and phasor approach lie in the difficulty of finding experimental systems and probes that meet the strict requirements. One key condition that must be met is that the decay constants in each phase τ_d and τ_o must be different enough that they are distinguishable on the universal circle, but not so different that the measurement loses sensitivity to the size-scales of interest.

Bibliography

[1] M. Digman, V. Caiolfa, M. Zamai, and E. Gratton. The phasor approach to flourescence lifetime imaging analysis. *Biophysical Journal*, 107:L14–L16, 2008.

[2] P. Garland and C. Moore. Phophorescence of protein-bound eosin and erythrosin. *Biochem. J.*, 183:561–572, 1979.

[3] N. James, J. Ross, M. Stefl, and D. Jameson. Applications of phasor plots to in vitro protein studies. *Anal. Biochem.*, 410:70–76, 2011.

[4] M. Stefl, N. James, J. Ross, and D. Jameson. Applications of phasors to in vitro time-resolved fluorescence measurments. *Anal. Biochem.*, 410:62–69, 2011.

[5] E. Yeramian and P. Claverie. Analysis of multiexponential functions without a hypothesis as to the number of components. *Nature*, 326:169–174, 1987.

Appendix E

Instrumentation

Part of my responsibilities in the Feigenson lab has been to maintain and repair the computers, electronics, and equipment. On occasion, experimental needs require us to modify existing equipment (such as fitting temperature ramping controllers to water baths) or build entirely new instruments from scratch. Two pieces of equipment that I have built and are routinely used in lab are the arbitrary function generator and a newly designed platinum wire chamber.

E.1 Arbitrary Function Generator and Oscilloscope

This arbitrary function generator was originally designed so that novel electroswelling routines could be tried that would allow the formation of GUVs in high salt environments. The user writes a simple script file (Vcycle.txt) that specifies the stages to be performed during the electroswelling process. These stages each have a specified time-frame (in seconds) during which a sine wave, square wave, or triangle wave can perform any combination of linear voltage-ramp and linear frequency-ramp. Once each stage is complete the voltage is held at ground indefinitely and a picture of the entire history of the ramping sequence is generated

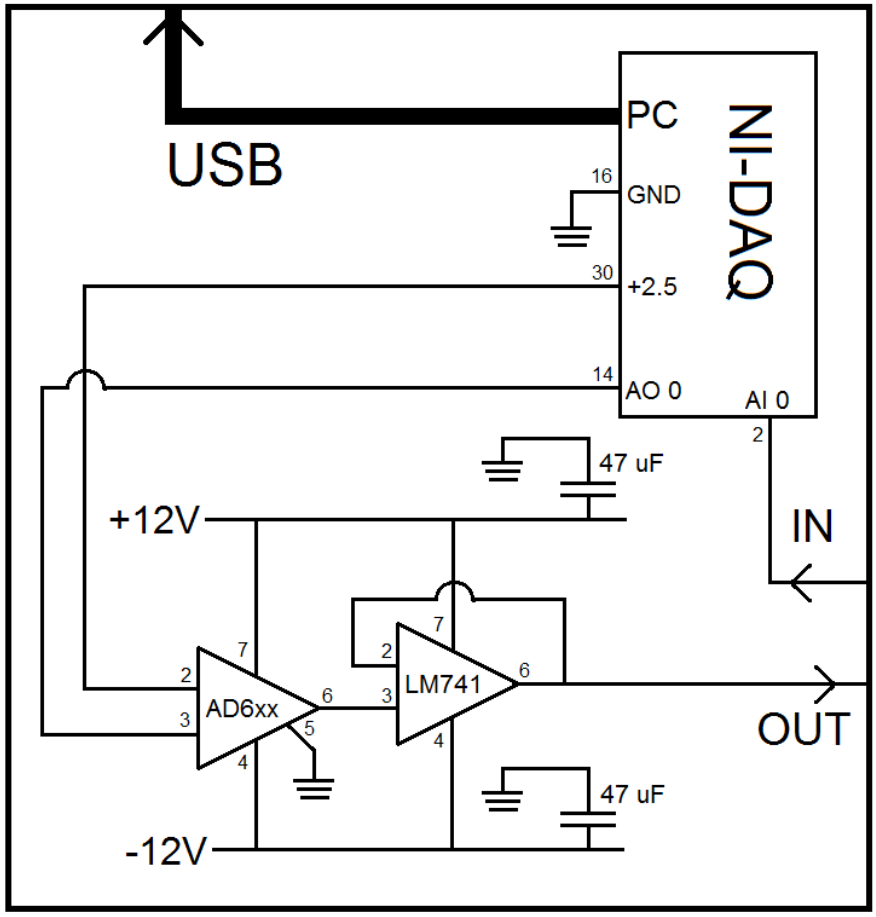


Figure E.1: Circuit diagram for the arbitrary function generator and oscilloscope. The output of the DAQ is split between ±12V rails, to produce a signal that is symmetric across ground, and then fed through a voltage follower before being output. The input is routed straight to the DAQ's analog-digital converter and sent to the PC for display.

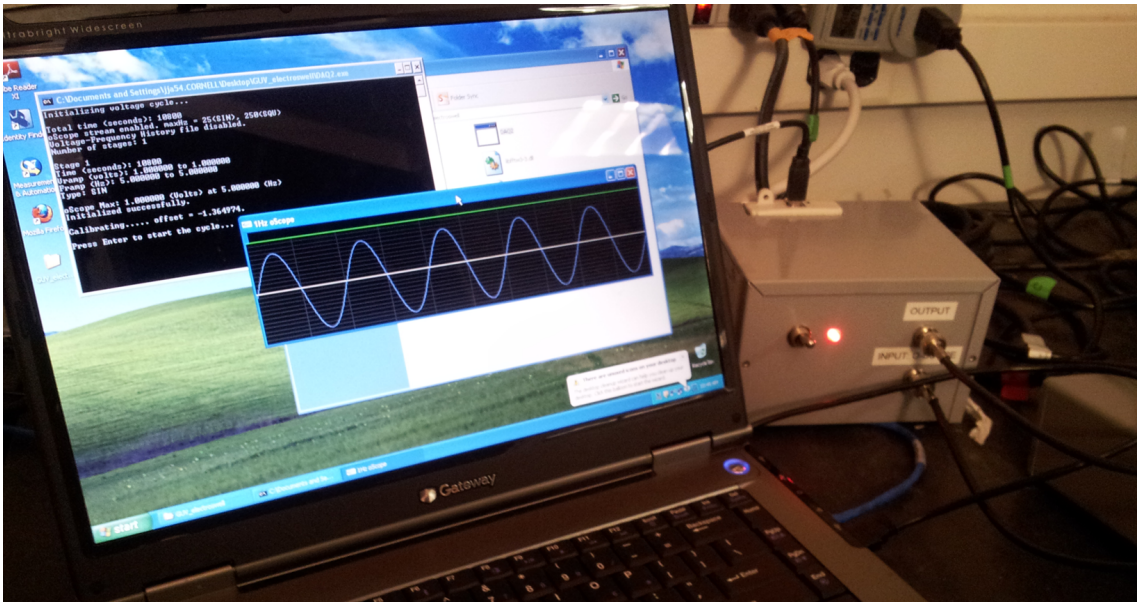


Figure E.2: Arbitrary function generator and oscilloscope showing a 5Hz, 2V peak-to-peak sine wave.

for quick error checking.

The hardware design is centered around a USB National Instruments Data Acquisition Card (NI DAQ). This piece of hardware interfaces with a custom piece of software that controls the ramp sequence and displays the measured voltage. A circuit diagram for the hardware portion is shown in figure E.1. The output of the DAQ is split between $\pm$12V rails to center the sine wave about ground. This signal is then buffered through a voltage follower and output to the samples. The input voltage feeds directly into the one of the analog-digital converters (ADC) of the DAQ and is sent to the software.

This instrument was never meant to be used as an oscilloscope. The voltage display is meant for users to verify that the ramp sequence is progressing as expected. When the oscilloscope our lab uses stopped working, the arbitrary function generator was re-purposed to function as an oscilloscope. By initializing the program with a dummy ramping sequence and routing the analog function generators output into the DAQ, the display software could be used as an oscilloscope. The functional range of this oscilloscope is very limited (maximum

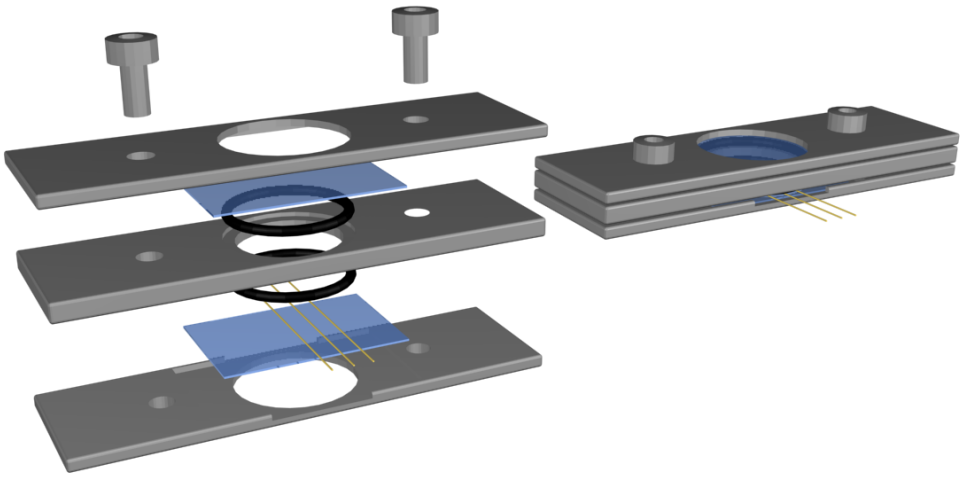


Figure E.3: Concept rendering of platinum wire chamber showing an exploded and assembled view.

observable frequency is 50Hz and maximum voltage is about 2.5V peak-to-peak), but this is sufficient for the parameters typically used in electroswelling. The display for a typical electroswelling run is shown in figure E.2. A read-me file is available in the program directory that gives all parameters and example use cases for new users.

E.2 Platinum Wire Chamber

One aspect of biomembranes that is difficult to reproduce in model systems is the preparation of GUVs with asymmetric leaflet compositions [2]. The protocol for making asymmetric GUVs is to start with symmetric GUVs with an outer leaflet composition. Then a solution containing β-cyclodextrin that is pre-loaded with inner leaflet lipids is flowed into the medium to exchange with lipids on the outer leaflet of the GUVs.

In order to realize this, one major problem must be overcome. When the medium is exchanged with free floating vesicles, even a gentle flow can push GUVs out of the field of view [3]. Using a platinum wire electrode for electroformation of GUVs may potentially solve

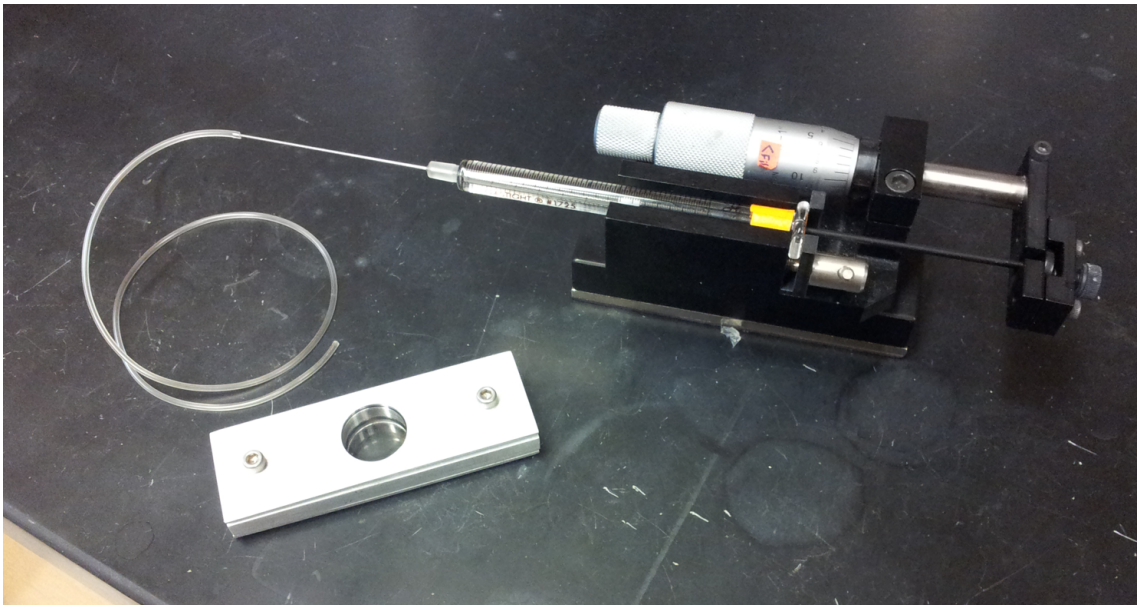


Figure E.4: Work in progress platinum wire chamber with micrometer-syringe to slowly exchange mediums in the chamber.

this problem. When GUVs are formed on platinum wire electrodes, the GUVs can remain tethered to the electrode through a thin tube of lipid. This small anchoring may be sufficient to hold the vesicles in the field of view during medium exchange.

The chamber I machined is sealed and sized to fit on the microscope stage so that we can observe the GUVs without having to harvest or perturb the samples. The same electroformation protocols will be used, just with platinum wire electrodes instead of ITO coated glass slides [1, 4]. A rendering of the final product is shown in figure E.3. In addition, we plan to use a micrometer syringe to change the medium at an arbitrarily slow pace. The current progress of the platinum wire chamber and the micrometer syringe is shown in figure E.4. This will further reduce the effects of fluid flow on the vesicles during exchange of medium.

Bibliography

[1] M. Angelova and D. Dimitrov. Liposome electroformation. *Farad. Discuss.*, 81:303–311, 1986.

[2] H. Cheng, Megha, and E. London. Preparation and properties of asymmetric vesicles that mimic cell membranes. *J. Biol. Chem.*, 284:6079–6092, 2009.

[3] N. Morales-Penningston, J. Wu, E. Farkas, S. Goh, T. Konyakhina, J. Zheng, W. Webb, and G. Feigenson. Guv preparation and imaging: Minimizing artifacts. *Biochim. Biophys. Acta.*, 1798:1324–1332, 2010.

[4] T. Politano, V. Froude, B. Jing, and Y. Zhu. Ac-electric field dependent electroformation of giant lipid vesicles. *Colloids Surf., B*, 79:75–82, 2010.

Appendix F

Programming Considerations

Here the organization and specifics of the Monte-Carlo simulation used in this study are explained. A short readme is presented for new users and a detailed listing of all data structures and file formats is given.

F.1 Overview and readme

This simulation is written in C++ and uses an object oriented approach to organize the various elements of the calculation. The visualization is handled by the OpenGL API using the GLUT library. For computational purposes only custom libraries are used with the exception of the standard C library. To avoid parameters with large exponents an internal set of units is used, $z(= 10^{-21}\text{J})$ is used for energy, μm is used for length, and e^- is used for charge. The random numbers used in the Monte Carlo portion of the simulation are generated using the routine 'Ranq1' defined in Numerical Recipes [1], which is seeded from the parameter file and has a period of $\approx 10^{19}$. A minimal working example (MWE) of C++ code is shown below.

```
#include <cstdio>
#include <cstdlib>
#include ‘‘simulation.h’’

int main()
{
        int i;
        Simulation sim; //initialize simulation class

        sim.ready(‘‘sPAR.csv’’); //reads parameter file

        for(i = 0; i < 100; ++i) //loop to perform many iterations
        {
                sim.run(); //perform a Monte Carlo sweep
        }

        sim.xportALL(); //export the data
        sim.release(); //release allocated memory

        return 0;
}
```

This MWE initializes the simulation from the parameter file (section F.3.1), performs 100 Monte Carlo sweeps, and then exports the data to disk. There are a whole host of methods that can be used with the **Simulation** and **Surface** class to access internal properties of the simulation. A practice that works nicely with the MWE described above is to initialize a reference to the surface, which allows the user to access and manipulate the surface shape directly using the **Surface** class methods. A modified MWE with this feature is shown below.

```
int main()
{
        int i;
```

```
        Simulation sim; //initialize simulation class
        Surface *psurface; //reference to a surface

        sim.ready(``sPAR.csv''); //reads in parameter file
        psurface = sim.pSurface(); //initialize the reference

        ...

        sim.xportALL(); //export the data
        psurface = 0; //de-reference the surface
        sim.release(); //release allocated memory

        return 0;
}
```

These examples and an understanding of the data structures used in the simulation should be sufficient to get started writing a C++ (or any other language that can reference C libraries) program that performs the lattice simulation.

Program Architecture

The simulation is written in an object oriented manner with the architecture shown in figure F.1. The **Simulation** class is the outermost structure initialized within the scope of **main()** and contains a single instance of the **Surface** class, an array of Nv (number of vertices) **GeoData** structures, a **Parameters** structure, and a **Scalings** structure. The **Surface** class contains an array of Nv **Vsimple** structures that store all of the information about the connectivity and shape of the surface. Each **Vsimple** structure has a linked list of indices, **neighbors**, which form a counter clock wise loop around the vertex and a **vf3** (3D vector) position of the vertex in space. The structures presented in figure F.1 all contain many additional entries and methods that are used during the simulation run time that are not shown. A more detailed listing of each data type and its usage is presented in section F.2.

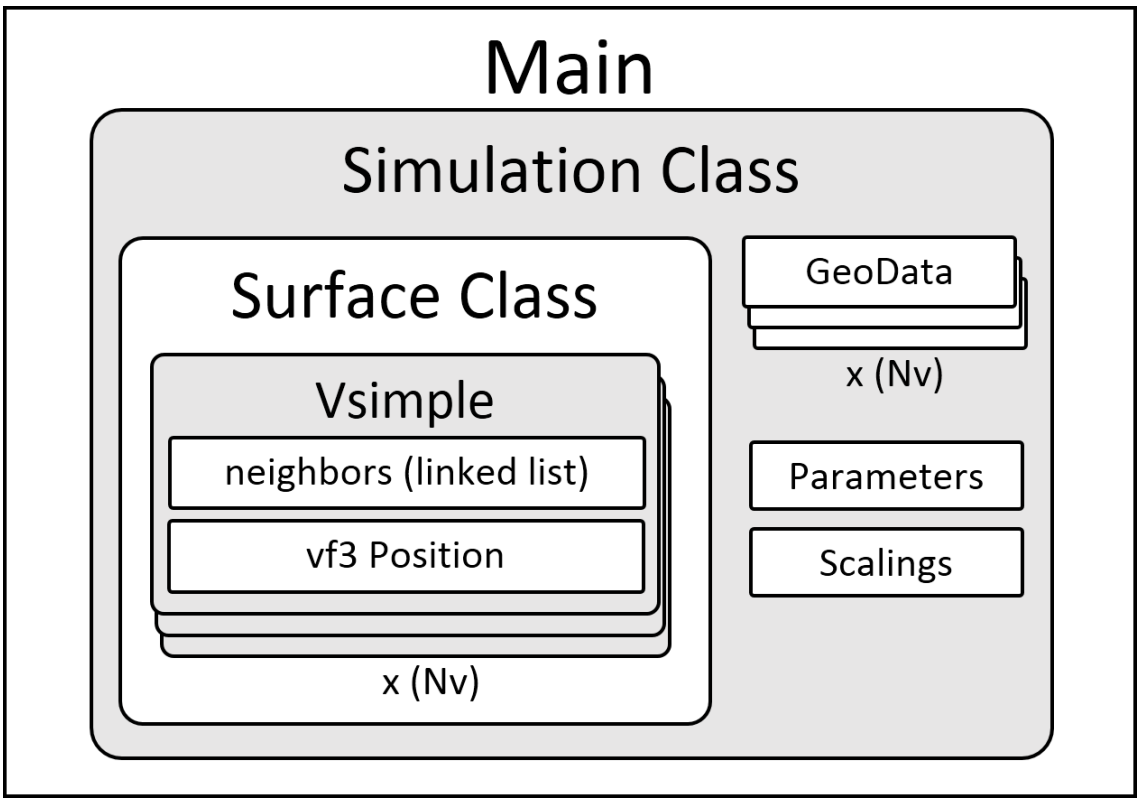


Figure F.1: Programming architecture of the simulation. The simulation contains a surface, which contains a list of vertices (Vsimple). The array of GeoData structures stores geometric information about the surface to expedite energy calculations.

Program Flow

The native implementation of the simulation includes a real-time 3D rendering of the current simulation status and some keyboard commands to manipulate the simulation and graphical representation. The program flow for this implementation is somewhat different from the previously presented MWE. The program enters at **main()** and begins initialization of all simulation data structures. The global pointer to a Simulation class is allocated and initialized, **pSim->ready("sPAR.csv")**, using the parameter file "sPAR.csv." This initialization routine reads in the energetics parameters and also initializes the Surface class of **pSim** by reading in the specified topology/constraint files. A global pointer to the surface, **pS = pSim->pSurface()**, is also initialized for easy access to the surface geometry by the OpenGL drawing routines. This series of routines fully prepares the simulation to run according to the user's flags, values, and surface topology specified in the parameter file.

Next the scope of the program is handed from **main()** to a timer function **update(int)** which runs once every 10 milliseconds indefinitely. Within the update routine the OpenGL **drawScene()** is called to render the current state of the simulation. Then the **if(!pause){pSim->run();}** routine is called, where **pause** is a global flag that is toggled with the '**p**' key (**pause** = true by default). The buffers are then swapped, **glutPostRedisplay()**, to update the window.

The **pSim->run()** call is the main iteration routine of the simulation. This function runs each of the various Monte Carlo exchanges; **anneal()** runs a single annealing step, **MCP(int)** runs the specified number of long range phase exchanges; **MCHG(int)** runs the specified number of vertex perturbations; **MCEF(int)** runs the specified number of edge flips. The routine then measures and outputs key status indicators such as the total area of the surface, the total volume of the surface, the integral of the square mean curvature (should be 4π for a sphere), the integrated Gaussian curvature (should be 4π for a sphere), and the

total phase boundary. The electrostatic field is updated for each vertex. Any specified measurements are appended to an output file and error flags from the various subroutines are handled. If all goes well **run()** returns 1. This returns scope to the **update()** loop which then preps for another iteration.

F.2 Data Structures

This section documents all of the data structures used in this simulation. Each data structures is implemented in its own header file that is included in the hierarchy of the program outlined in figure F.1. In the following section(s) member variables are shown in blue, methods and functions are shown in red, and files and file entries are shown in green.

F.2.1 vf3 structure

The vf3 data structure is an organizational container for three dimensional vectors. It allows easy mathematical manipulation of vector quantities through access to its individual components and a small suite of operators/methods. All vectors in the simulation are stored in this way.

```
struct vf3
{
        float X,Y,Z;
        vf3(float x, float y, float z);
        void zero();
        float length();
        void normalize();
        void operator- ();
        void operator+= (vf3 B);
        void operator-= (vf3 B);
        void operator*= (float a);
        void operator= (vf3 B);
```

```
        vf3 operator+ (vf3 B);
        vf3 operator- (vf3 B);
        vf3 operator* (float a);
        vf3 operator/ (float a);
        float operator* (vf3 B);
        float operator< (vf3 B);
        float operator<< (vf3 B);
        vf3 operator& (vf3 B);
};
extern inline float areaV2(vf3 A, vf3 B);
extern inline float volumeV3(vf3 A, vf3 B, vf3 C);
vf3 normalV2(vf3 A, vf3 B);
```

Members

float X,Y,Z: Vector components.

Methods

vf3(float x, float y, float z): Constructor that initializes the vector.

(vf3).zero(): Sets all components of the vector to zero.

(vf3).length(): Returns the euclidean length of the vector.

(vf3).normalize(): Normalizes the vector.

-(vf3): Negation operator.

+=(vf3): Unitary addition.

-=(vf3): Unitary subtraction.

***=(vf3):** Unitary scalar multiplication.

=(vf3): Assignment operator.

(vf3)+(vf3): Binary addition.

(vf3)-(vf3): Binary subtraction.

(vf3)*(float): Binary scalar multiplication.

(vf3)/(float): Binary scalar division.

(vf3)*(vf3): Dot product.

(vf3)&(vf3): Cross product.

(vf3)<(vf3): The raw angle between two vectors, returns a number between 0 and π.

(vf3)<<(vf3): The smallest angle between two vectors. This operator treats the vectors as lines extending infinitely in both directions and returns the smallest angle between them, always between 0 and $\pi/2$.

areaV2(vf3,vf3): Returns the area of the triangle defined by two vectors.

volumeV3(vf3,vf3,vf3): Returns the volume of a pyramid defined by three vectors.

normalV2(vf3,vf3): Returns a normalized vector orthogonal to the two input vectors.

F.2.2 Vsimple structure

The Vsimple data type is a means of storing all information relevant to a single vertex of a surface. It keeps track of the position of a given vertex and its connectivity with the rest of the lattice. It has no methods, but is manipulated through the methods of a parent data structure, the 'Surface' class.

```
struct Vsimple
{
        Chain<unsigned int> neighbors;
        vf3 position;
        unsigned int Nb,cut;
};
```

Members

Chain<unsigned int> neighbors: A circularly linked list of the indices of vertices that this vertex is connected to. The neighbors are ordered to form a counter clockwise loop around the parent vertex. The 'Chain' data type contains an internal iterator that allows traversal of the linked list. The reason a linked list is used instead of a simple array is the ease with which a neighbor can be deleted/inserted, allowing topological changes to the lattice with minimal memory management. A special value, 'BREAK' $\equiv$ 0xFFFFFFFF, is used to indicate a gap in the neighbor array at the edge of a surface.

vf3 position: Cartesian coordinates of the vertex.

unsigned int Nb: Number of neighbors this vertex has.

unsigned int cut: Flag that indicates if this vertex is at the edge of a surface (0 - no, 1 - yes). This is used to quickly assess which routines must be used to update this vertex.

F.2.3 GeoData structure

The GeoData structure stores all of the geometric data associated with a given vertex. This is to minimize the amount of redundant calculations that need to be done. The structure is populated when the simulation is initialized and updated only when the geometric properties

of the vertex have changed (area, normal, curvature, etc...). This structure has no methods and is manipulated with the methods of a parent data structure, the 'Simulation' class.

```
struct GeoData
{
        float area;
        float volume;
        float H;
        float G;
        float boundary;
        unsigned int Nb;
        unsigned int Npartner;
        vf3 normal,E;
        float edges[MAX_NB];
        float oedges[MAX_NB];
        unsigned int partners[MAX_PARTNER];
};
```

Members

float area: Area of this vertex (defined as 1/3 the area of the adjacent faces).

float volume: Volume of this vertex (defined as 1/3 volume of the cone formed by the adjacent faces and the origin).

float H: Mean curvature of this vertex.

float G: Gaussian curvature of this vertex.

float boundary: Total phase boundary associated with this vertex.

unsigned int Nb: Number of neighbors this vertex has.

unsigned int Npartner: Number of partners this vertex has. Partners are the vertices that this vertex can interact with electrostatically (those within a cut-off distance).

vf3 normal: Normal vector of the surface at this vertex (defined as the area-weighted sum of the normals of the adjacent faces).

vf3 E: Electric field at this vertex as defined using equation 3.5.

float edges[MAX_NB]: Array of lengths of the adjacent edges. The array size MAX_NB is defined as 10.

float oedges[MAX_NB]: Array of lengths of the arcs orthogonal to the edges. The arc is defined as the two line segments joining the center of the previous face to the center of the edge and the center of the edge to the center of the next face. This useful for quickly calculating portions of the phase boundary.

unsigned int partners[MAX_PARTNER]: Array of indices of partner vertices. The array size MAX_PARTNER is defined as 10,000.

F.2.4 Parameters structure

The parameters data structure is a storage container for the values read in from the parameter file. It contains all of the energetic values that define the model as well as run-time values/flags that configure the simulation to run in a certain way. This data structure has no methods.

```
struct Parameters
{
        float g,k0,k1,kG0,kG1,C0,C1,Cp,u0,u1,ap0,ap1,kdebye;
        float kT,kTstart,kTstop,R,P,wE,wA,wR,rAG,rVG;
        unsigned int seed, kTsweeps;
        unsigned char fEp,fEH,fEG,fEu,fEA,fEe,fEn,fEglass,fGA,fGV;
        unsigned char fMCP,fMCHG,fMCEF,fgdescent,fanneal;
};
```

Members

float g: Scale invariant line tension (equivalent to u defined in section 3.4.1). This is converted to the units of z/μm internally at initialization.

float k0,k1: Mean curvature bending modulus (in units of z) for L_d and L_o phase respectively.

float kG0,kG1: Gaussian curvature bending modulus (in units of z) for L_d and L_o phase respectively.

float C0,C1: Spontaneous curvature (in units of μm^{-1}) for L_d and L_o phase respectively.

float Cp: Spontaneous curvature (in units of μm^{-1}) for generalized protein field.

float u0,u1: Dipole density (in units of $\text{e}^-/\mu\text{m}$) for L_d and L_o phase respectively.

float ap0,ap1: Coupling of generalized protein field (in units of z/protein) for L_d and L_o phase respectively.

float kdebye: Decay length of the electrostatic field.

float kT: Current temperature (in units of z). Initialized to kTstart.

float kTstart,kTstop: Starting and ending temperature (in units of z) of the annealing routine.

float R: Simulation size scale (in units of μm).

float P: Area fraction of L_o phase.

float wE: Wiggle room of edges (fraction, ≈ 0.3).

float wA: Wiggle room of area of faces (fraction, ≈ 0.3).

float wR: Distance to move a vertex in a single exchange (as a fraction of R, ≈ 0.001).

float rAG,rVG: Area/Volume modulators within the global area/volume constraint. Typically left as 1.0 (< 1.0 indicates contraction of area/deflation of the vesicle; > 1.0 indicates dilation of area/swelling of the vesicle).

unsigned int seed: Seed used to initialize the random number generator.

unsigned int kTsweeps: Number of Monte Carlo steps to cool from kTstart to kTstop.

unsigned char fEp: Flag for line tension term in the energy functional.

unsigned char fEH: Flag for mean curvature term in the energy functional.

unsigned char fEG: Flag for Gaussian curvature term in the energy functional.

unsigned char fEu: Flag for electrostatics term in the energy functional.

unsigned char fEA: Flag for local area constraint.

unsigned char fEe: Flag for local edge constraint.

unsigned char fEn: Flag for normals constraint.

unsigned char fEglass: Flag for glass support energy term.

unsigned char fGA: Flag for global area constraint.

unsigned char fGV: Flag for global volume constraint.

unsigned char fMCP: Flag to perform Monte Carlo phase exchanges. At each iteration fMCP*Nv/10 exchanges will be attempted.

unsigned char fMCHG: Flag to perform Monte Carlo vertex perturbations. At each iteration fMCHG*Nv/10 movements will be attempted.

unsigned char fMCEF: Flag to perform Monte Carlo edge flips. At each iteration fMCEF*Nv/10 flips will be attempted.

unsigned char fgdescent: Flag to perform gradient descent on each vertex (not currently implemented).

unsigned char fanneal: Flag to perform annealing routine. If set to zero, temperature will remain at starting temperature.

F.2.5 Scalings structure

The scalings struct contains information used by the OpenGL drawing routines to properly scale and color the various fields that are rendered on the screen. This data structure has no methods.

```
struct Scalings
{
        float Emin,Emax,Hmin,Hmax,Gmin,Gmax;
};
```

Members

float Emin,Emax: Minimum and Maximum magnitudes of the electric fields.

float Hmin,Hmax: Minimum and Maximum mean curvature values.

float Gmin,Gmax: Minimum and Maximum Gaussian curvature values.

F.2.6 Surface class

The surface class contains information about the surface as a whole. It is initialized via a topology file that is used to populate an array of 'Vsimple' structures with the position and neighbor arrays of each vertex. This data structure has many methods that are used to read and manipulate the vertices of the lattice.

```
class Surface
{
private:
        Vsimple *vertices;
        unsigned int Nv,Ne,Nf;
public:
        void ready(const char *fname);
        void ready(unsigned int numV, unsigned int numE, unsigned int numF);
        void xportSurface(const char *outname);
        void release();
        unsigned int getNv();
        unsigned int getNe();
        unsigned int getNf();
        unsigned int getNb(unsigned int v);
        unsigned int getCut(unsigned int v);
        unsigned int getVneighbor(unsigned int v);
        unsigned int getVneighborF(unsigned int v);
        unsigned int findVneighbor(unsigned int v, unsigned int vN);
        void insertVneighborF(unsigned int v, unsigned int vN);
        void deleteVneighborB(unsigned int v);
        vf3 getVposition(unsigned int v);
        void setVposition(unsigned int v, vf3 vtemp);
        void goPrev(unsigned int v);
        void goNext(unsigned int v);
};
```

Members

Vsimple *vertices: Pointer to an array of Vsimple objects. This array is populated during initialization from a *.TOP file and contains the entire surface that will be used in the simulation.

unsigned int Nv,Ne,Nf: Number of vertices, number of edges, and number of faces.

Methods

ready(const char *fname): Initialization routine that takes the name of a *.TOP file to be read in. This routine populates the positions and neighbor arrays of each element of the vertices array.

ready(unsigned int numV, unsigned int numE, unsigned int numF): Initializes a dummy surface for testing.

xportSurface(const char *outname): Exports a topology file for the current state of the surface.

release(): Releases all memory used by this surface.

getNv(): Returns the number of vertices on this surface.

getNe(): Returns the number of edges on this surface.

getNf(): Returns the number of faces on this surface.

getNb(unsigned int v): Returns the number of neighbors that vertex v has.

getCut(unsigned int v): Returns the value of 'cut' for the given vertex.

getVneighbor(unsigned int v): Returns the index of the neighbor currently pointed to by the iterator of the 'neighbors' chain of the given vertex.

getVneighborF(unsigned int v): Returns the index of the neighbor currently pointed to by the iterator of the 'neighbors' chain of the given vertex **and** moves the iterator forward one entry.

findVneighbor(unsigned int v, unsigned int vN): Searches the neighbor array of the vertex v for the entry vN. Returns 0 if not found and 1 if it is found. If it is found, it sets the iterator to the position of vN.

insertVneighborF(unsigned int v, unsigned int vN): Inserts the neighbor vN into the neighbor array ahead of the current iterator position. Iterator now points to the new entry.

deleteVneighborB(unsigned int v): Deletes the neighbor entry that is currently pointed to by the iterator. Iterator now points to the preceding entry.

getVposition(unsigned int v): Returns the vector position of this vertex.

setVposition(unsigned int v, vf3 vtemp): Sets the position of vertex v to the vector vtemp.

goPrev(unsigned int v): Moves the neighbor iterator of vertex v back one entry.

goNext(unsigned int v): Moves the neighbor iterator of vertex v forward one entry.

F.2.7 Simulation class

The 'Simulation' class is the apex structure in the hierarchy of this implementation. It is defined as a singleton that contains all of the information necessary to represent and run the model. The methods of this class perform the Monte Carlo iterations that relax the lattice to a minimal free energy configuration. This class is the most developed structure of the

simulation and all of it's methods are called directly from 'main' or OpenGL draw functions (to poll the model state).

```
class Simulation
{
private:
        Parameters rawparam,param;
        Scalings scalings;
        Surface *surface;
        GeoData *geodata;
        unsigned int Nv,Ne,Nf;
        unsigned int GoHeatCap;
        char fnameTOP[256];
        char fnameCON[256];
        char fnameOTOP[256];
        char fnameOPAR[256];
        char fnameOCON[256];
        char fnameCORR[256];
        char fnameENERGY[256];
        stdRan *random;
        unsigned int *Lp;
        float *Prot1;
        float cE;
        float Aglobal,A0,Vglobal,V0;
        unsigned int sweeper;
        float E_local(unsigned int v);
        float E_global();
        float EnergyTotal();
        void fillGeoPartners(unsigned int v);
        void boundaryGeoData(unsigned int v);
        void EGeoData(unsigned int v);
        void fillGeoData(unsigned int v);
        void updateScale();
        unsigned int MCP(unsigned int iter);
        unsigned int MCPNN(unsigned int iter);
        unsigned int MCPROT(unsigned int iter);
        unsigned int MCHG(unsigned int iter);
        unsigned int MCEF(unsigned int iter);
        void anneal();
        unsigned int gdescent(unsigned int iter);
```

```
        void xportTOP(const char* outname);
        void xportPAR(const char* outname);
        void xportCON(const char* outname);
public:
        void ready(const char* fname);
        void xportALL();
        int run();
        void release();
        void CalcCorrelationFunction();
        void toggleHC(){GoHeatCap = (GoHeatCap+1)%2;}
        float CalcNNcorrelation();
        Surface* pSurface();
        Scalings* pScalings();
        unsigned int getLp(unsigned int v);
        float getProt1(unsigned int v);
        float getH(unsigned int v);
        float getG(unsigned int v);
        vf3 getnorm(unsigned int v);
        vf3 getE(unsigned int v);
        float getR();
};
```

Members

Parameters rawparam,param: Parameter structures to hold the values describing the model. The parameters are read directly from the file into the rawparam structure and then copied to the param structure. During the simulation run time all parameters are polled from param. The reason for a copy is that parameters can be changed within the simulation, but if we export the parameters we would like to recover the initial parameters fed into the simulation. Thus the export routine reads values from rawparam.

Scalings scalings: Scalings used for OpenGL draw routines. This structure is updated once every Run() call to keep the values up to date.

Surface *surface: Pointer to a Surface class.

GeoData *geodata: Pointer to an array of GeoData structures (one for each vertex on the surface).

unsigned int Nv,Ne,Nf: Number of vertices, number of edges, number of faces.

unsigned int GoHeatCap: A general purpose toggle flag for various measurements.

char fnameTOP[256]: Name of topology (*.TOP) file to be read in.

char fnameCON[256]: Name of constraint (*.CON) file to be read in.

char fnameOTOP[256]: Name of topology (*.TOP) file to be exported.

char fnameOPAR[256]: Name of parameter (*.csv) file to be exported.

char fnameOCON[256]: Name of constraint (*.CON) file to be exported.

char fnameCORR[256]: Name of correlation function file to be exported.

char fnameENERGY[256]: Name of energy measurement file to be exported (used for heat capacity).

stdRan *random: Pointer to random number generator struct.

unsigned int *Lp: Array of Nv integers that represent the phase field ($0 = L_d$, $1 = L_o$).

float *Prot1: Array of Nv floats that represent the protein concentration field.

float cE: Average length of an edge between two vertices.

float Aglobal,Vglobal: Current value of the global area(volume).

float A0,V0: Initial values of the global area(volume).

unsigned int sweeper: General purpose program counter to keep track of number of iterations elapsed.

Private Methods

E_local(unsigned int v): Returns the local energy of vertex v.

E_global(): Returns the energy of the global constraints.

EnergyTotal(): Returns the total energy of the entire lattice.

fillGeoPartners(unsigned int v): Fills the partners array of the GeoData entry for vertex v.

boundaryGeoData(unsigned int v): Fills the boundary and oedges array of the GeoData entry for vertex v.

EGeoData(unsigned int v): Computes the electric field vector of the GeoData entry for vertex v.

fillGeoData(unsigned int v): Fills all of the data fields in the GeoData entry for vertex v.

updateScale(): Updates the values in the scalings struct.

MCP(unsigned int iter): Performs 'iter' long range phase exchanges. Returns the number of successful phase exchanges performed.

MCPNN(unsigned int iter): Performs 'iter' nearest neighbor phase exchanges. Returns the number of successful phase exchanges performed.

MCPROT(unsigned int iter): Performs 'iter' protein field exchanges. Returns the number of successful exchanges performed.

MCHG(unsigned int iter): Performs 'iter' vertex perturbations. Returns the number of successful vertex moves performed.

MCEF(unsigned int iter): Performs 'iter' edge flip attempts. Returns the number of successful edge flips performed.

anneal(): Performs one annealing step (decrements param.kT by (param.kTstart - param.kTstop)/param.kTsweeps).

gdescent(unsigned int iter): Perform 'iter' gradient descent steps. Returns number of gradient descent steps performed.

xportTOP(const char* outname): Exports the topology file for the current state of the simulation.

xportPAR(const char* outname): Exports the parameter file for this simulation.

xportCON(const char* outname): Exports the constraint file for the current state of the simulation.

Public Methods

ready(const char* fname): Reads in the parameter file 'fname' and initializes all arrays and memory in the Simulation class.

xportALL(): Exports all three files necessary to specify the simulation state; topology (*.TOP), constraint (*.CON), and parameter (*.csv).

run(): The master command. Each call to run() performs a single Monte Carlo sweep of the lattice. Each sweep consists of performing exchanges, logging relevant observables, printing status information to the console, and updating any quantities that may have

changed over the course of the step. The run() method is called as fast as possible in a tight update loop.

release(): Releases all memory used by this simulation.

CalcCorrelationFunction(): Calculates the two point correlation function for the phase field and exports it to a file.

toggleHC(): Toggles the value of GoHeatCap between 0 and 1.

CalcNNcorrelation(): Returns the probability of two adjacent vertices having the same phase.

pSurface(): Returns a pointer to the surface class. Used for OpenGL drawing routines to have direct access to the vertex positions.

pScalings(): Returns a pointer to the scalings structures.

getLp(unsigned int v): Returns the phase of vertex v.

getProt1(unsigned int v): Returns the protein field of vertex v.

getH(unsigned int v): Returns the mean curvature of vertex v.

getG(unsigned int v): Returns the Gaussian curvature of vertex v.

getnorm(unsigned int v): Returns the normal vector of the surface at the vertex v.

getE(unsigned int v): Returns the electric field vector at the vertex v.

getR(): Returns the size scale parameter, param.R.

F.3 File Formats

This simulation uses three files to define the simulation state. At initialization these files are read into the simulation and used to construct the surface, populate the parameter structures, and prepare the energy functional. When the user chooses (or a user defined condition is met), the key combination '**shift+X**' will output the three files that represent the current state of the simulation to the working directory. These files can subsequently be used for analysis or to restart the simulation from the same point.

F.3.1 Parameter file

The parameter file is **sPAR.csv** (comma separated values) format. This is a human readable and editable file format. Any text editor or spreadsheet program (such as Excel) can be used to edit the parameters. At run time these values are imported into the rawparam and param data structure of the **Simulation** singleton. An example of the **sPAR.csv** file is shown below.

Topology File,s10000.TOP

Constraint File,-

OutputTOP,testTOP.TOP

OutputPAR,testPAR.csv

OutputCON,testCON.CON

seed,0

fEp,1

fEH,1

fEG,1

fEu,0

fEA,0

fEe,1

fEn,0

fEglass,0

fGA,1

fGV,0

fMCP,5

fMCHG,1

fMCEF,0

fgdescent,0

fanneal,1

k0,200

k1,1000

kG0,-200

kG1,-1000

C0,0

C1,0

u0,133

u1,309

kdebye,0.002

g,2.8

R,0.1

P,0.5

rAG,1

rVG,1

wE,0.3

wR,0.001

kTstart,4.0

kTstop,4.0

kTsweeps,10000

Most of these parameters have a one-to-one correspondence to the parameters structure (section F.2.4). The constraint file entry may contain a string that specifies a *.CON file in the working directory or may be just a '-' (as shown above). The dash signals the program to populate the vertices with L_o and L_d phase at random, but with proper area fraction as specified by P.

F.3.2 Topology file

The topology file contains all geometric and topological information about the surface. It has a *.TOP file extension and is in a binary (not human readable) format. The structure of a *.TOP file is shown below.

[**int** number of vertices] [**int** number of edges] [**int** number of faces]

[**int** index] [**float** X coordinate] [**float** Y coordinate] [**float** Z coordinate]
[**int** number of neighbors] $\underbrace{[\textbf{int}\ \text{neighbor one}]...[\textbf{int}\ \text{neighbor ten}]}_{\times 10}$ $\underbrace{[\textbf{int}\ \text{NULL}]...[\textbf{int}\ \text{NULL}]}_{\times 20}$

$\vdots$

$\times$ number of vertices

$\vdots$

All of the values in this file are 32 bit data types. The first three entries indicate the number of vertices, edges, and faces on the surface. After that, there is a repeated entry

pattern for each vertex that includes the index of the vertex (0 to (Nv-1)), its coordinates in space, the number of neighbors it has, and the indices of its neighbors. The trailing twenty zeros are placeholders for older versions of the format to maintain backwards compatibility. In order to facilitate vertices that may be on the edge of a surface a special value is used to indicate the gap in the neighbor array, 'BREAK' ≡ 0xFFFFFFFF. This value counts as a neighbor and thus is included in the number of neighbors entry. The entries in the neighbors must for a counter clock-wise loop around the parent vertex.

F.3.3 Constraint file

The constraint file is a human readable file with file extension *.CON. This file contains information about the fields at each vertex (at this point just the phase field). An example constraint file is shown below.

Nv:10000

0: 1

1: 0

2: 0

⋮

Nv-3: 0

Nv-2: 1

Nv-1: 1

The first line of file indicates the number of vertices followed by an entry for each vertex of the form [**int** index]:[**int** phase field]. The phase field takes on one of two values, 0 for L_d and 1 for L_o.

Bibliography

[1] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes, The Art of Scientific Computing, Third Edition*, volume 1. Cambridge University Press, 2007.